

UNIVERSITY
OF IOANNINA

UNIVERSITY OF IOANNINA

DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

M.Sc. Program in Informatics and Networks

MASTER'S THESIS

UAV-based Advanced Geolocation Estimation with Integrated Computer Vision Methods

---

Dimitrios Mpouziotas (AM: 195)

Supervisor: Petros Karvelis
Associate Professor

Arta, Greece, 47100
Publish Date: November 2, 2026

Προσδιορισμός Γεωγραφικών Συντεταγμένων Αντικειμένων Από Βίντεο Drone Με Χρήση Υπολογιστικής Όρασης

Approved by the Three-Member Examination Committee
Arta, Greece, October, 22, 2026

EVALUATION COMMITTEE

1. Primary Supervisor

Petros Karvelis
Associate Professor

2. Committee Member

Stylios Chrysostomos
Professor

3. Committee Member

Ioannis Tsoulos
Professor

© Dimitris Bouziotas, 2026
All rights reserved.

Declaration of Non-Plagiarism

I hereby solemnly declare, being aware of the penalties provided for under Law 2121/1993 on Intellectual Property, that this master's thesis is entirely the result of my own research, does not constitute a product of plagiarism, and has not been commissioned from third parties. All sources used in its preparation, regardless of their type, format, or origin, are included in the bibliography.

Dimitris Bouziotas

Signature



ACKNOWLEDGEMENTS

This work was developed with the guidance of my supervisors Petros Karvelis and Chrysostomos Stylios and was conducted at the Laboratory of Knowledge and Intelligent Computing (KIC). I would like to express my sincere gratitude and appreciation during this whole journey, since they provided me with the right tools, knowledge and education to achieve great lengths in my academic career. I am deeply thankful of their patience and support allowing me to complete this research along with several others. I would also like to express my gratitude and appreciation to my family for constantly encouraging and motivating me to pursue great lengths I never imagined myself to be in. My professors funded and allowed me to promote this work in the foreign conference Smart World Congress (SWC) 2026, Italy Rende, where I successfully presented an innovative application with real-world use-cases, entitled Probabilistic MultiView Geolocation (PMV-Geo) that was originated from this study. Part of this research is carried out within the framework of the CRADLES project «Creating Resilient Areas to Develop Lifecycles and Ecosystem Services» funded under the Interreg VIB IPA Adriatic Ionian programme co-funded by the European Union and National funds (project code:IPA-ADRION00389).

ABSTRACT

Traditional methods to accurately localize objects such as vehicles, pedestrians, or coastal vegetation over large geographic areas often rely on physical hardware implants like GPS tags, which are costly, labor-intensive, and restricted to pre-selected targets. To overcome these limitations, this thesis presents an end-to-end framework that captures high-resolution imagery and video footage from Unmanned Aerial Vehicles (UAVs) and leverages state-of-the-art computer vision architectures to automate target detection, semantic segmentation, multi-object tracking, and geolocation estimation. The system framework integrates onboard UAV sensor telemetry including active gimbal pitch and yaw angles, GNSS positioning, and relative flight altitude to project 2D image-plane coordinates into 3D global geodetic coordinates (WGS-84). This transformation mathematically models camera intrinsic parameters, asymmetric 16:9 sensor cropping principal point displacements, 5-parameter Brown-Conrady lens distortion, 3D gimbal rotation matrices, and altitude-scaled ray-to-ground intersections under a locally flat ground plane baseline ($z = 0$). The operational validity of the pipeline is evaluated across two distinct real-world applications: first, an automated Probabilistic Multi-View Geolocation (PMV-Geo) framework coupling deep semantic segmentation (YOLO26x-sem) with Gaussian-weighted rasterization across accumulated probability and density matrices to map coastal salt marsh canopies in the Amvrakikos Gulf (GSD $\approx$ 4.09 cm/px) without topological polyline fragmentation, evaluated via Symmetric Cross-Correlation Evaluation (SCCE) using Zero-Mean Normalized Cross-Correlation (ZNCC) to achieve 2.90 – 3.17 m mean shoreline precision across multi-year temporal shifts; second, a real-time target detection and tracking pipeline integrating edge-optimized YOLO26 architectures with BotSORT tracking for aerial vehicle and pedestrian localization, utilizing a Within-Bounds (WB) geometric filter to eliminate centroid drift induced by partial boundary truncation along frame edges. Experimental evaluations under controlled flight profiles achieve a track-level median Root Mean Square Error (RMSE) down to 0.498 m, a 50% Circular Error Probable (CEP_{50}) of 0.395 m, and a 95% operational bound (CEP_{95}) of 0.698 m, providing a highly accurate, cost-effective, and scalable solution for ecological monitoring, urban surveillance, and search and rescue operations.

Keywords: Geolocation Estimation, Computer Vision, Unmanned Aerial Vehicle (UAV), Vegetation Mapping, Target Tracking, Geospatial Data Processing, PMV-Geo, Semantic Segmentation, Multi-Object Tracking, Satellite Imagery

ΠΕΡΙΛΗΨΗ

Οι παραδοσιακές μέθοδοι για τον ακριβή εντοπισμό αντικειμένων, όπως οχήματα, πεζοί ή παράκτια βλάστηση σε μεγάλες γεωγραφικές περιοχές, συχνά βασίζονται σε φυσικές ενσωματώσεις υλικού όπως ετικέτες GPS, οι οποίες είναι δαπανηρές, απαιτούν μεγάλη ένταση εργασίας και περιορίζονται σε προεπιλεγμένους στόχους. Για την αντιμετώπιση αυτών των περιορισμών, η παρούσα διατριβή παρουσιάζει ένα ολοκληρωμένο πλαίσιο (end-to-end) που συλλέγει εικόνες και βίντεο υψηλής ανάλυσης από Μη Επανδρωμένα Αεροχήματα (UAVs) και αξιοποιεί αρχιτεκτονικές υπολογιστικής όρασης αιχμής για την αυτοματοποίηση του εντοπισμού στόχων, της σημασιολογικής κατάτμησης, της παρακολούθησης πολλαπλών αντικειμένων και της εκτίμησης γεωγραφικού εντοπισμού. Το πλαίσιο του συστήματος ενσωματώνει τηλεμετρία αισθητήρων του UAV, συμπεριλαμβανομένων των ενεργών γωνιών πρόνευσης (pitch) και εκτροπής (yaw) του gimbal, της τοποθέτησης GNSS και του σχετικού ύψους πτήσης, για την προβολή 2D συντεταγμένων επιπέδου εικόνας σε 3D παγκόσμιες γεωδαιτικές συντεταγμένες (WGS-84). Αυτός ο μετασχηματισμός μοντελοποιεί μαθηματικά τις εσωτερικές παραμέτρους της κάμερας, τις μετατοπίσεις του κύριου σημείου λόγω ασύμμετρης αποκοπής αισθητήρα 16:9, παραμορφώσεις φακού 5 παραμέτρων κατά Brown-Conrady, πίνακες περιστροφής 3D gimbal και τομές ακτίνας-εδάφους κλιμακούμενες με το ύψος υπό την παραδοχή τοπικά επιπέδου εδάφους ($z = 0$). Η επιχειρησιακή εγκυρότητα της ροής εργασιών αξιολογείται σε δύο διαφορετικές εφαρμογές πραγματικού κόσμου: πρώτον, ένα αυτοματοποιημένο πλαίσιο Πιθανοτικής Γεωγραφικής Εντοπιστικότητας Πολλαπλών Όψεων (PMV-Geo) που συνδυάζει βαθιά σημασιολογική κατάτμηση (YOLO26x-sem) με καννάβωση (rasterization) στάθμισης Gauss σε συσσωρευμένους πίνακες πιθανότητας και πυκνότητας για τη χαρτογράφηση της παράκτιας βλάστησης αλμυρών βάλτων στον Αμβρακικό Κόλπο (GSD ≈ 4.09 cm/px) χωρίς τοπολογικό κατακερματισμό πολυγραμμών, αξιολογημένο μέσω Συμμετρικής Αξιολόγησης Διασταυρούμενης Συσχέτισης (SCCE) με χρήση Διασταυρούμενης Συσχέτισης Μηδενικού Μέσου Όρου Κανονικοποιημένης (ZNCC) επιτυγχάνοντας μέση ακρίβεια ακτογραμμής 2.90 – 3.17 m σε πολυετείς χρονικές μετατοπίσεις· δεύτερον, μια ροή εργασιών εντοπισμού και παρακολούθησης στόχων σε πραγματικό χρόνο που ενσωματώνει αρχιτεκτονικές YOLO26 βελτιστοποιημένες για άκρα (edge) με αλγόριθμο παρακολούθησης BotSORT για τον εντοπισμό εναέριων οχημάτων και πεζών, χρησιμοποιώντας ένα γεωμετρικό φίλτρο εντός ορίων (Within-Bounds – WB) για την εξάλειψη της ολίσθησης κεντροειδούς που προκαλείται από μερική αποκοπή ορίων κατά μήκος των άκρων του πλαισίου. Οι πειραματικές αξιολογήσεις υπό ελεγχόμενα προφίλ πτήσης επιτυγχάνουν διάμεσο Ριζικό Μέσο Τετραγωνικό Σφάλμα (RMSE) σε επίπεδο τροχιάς έως 0.498 m, Πιθανό Κυκλικό Σφάλμα 50% (CEP_{50}) 0.395 m, και επιχειρησιακό όριο 95% (CEP_{95}) 0.698 m, προσφέροντας μια εξαιρετικά ακριβή, οικονομικά αποδοτική και κλιμακώσιμη λύση για οικολογική παρακολούθηση, αστική επιτήρηση και επιχειρήσεις έρευνας και διάσωσης.

Λέξεις-κλειδιά: Εκτίμηση Γεωγραφικού Εντοπισμού, Υπολογιστική Όραση, Μη Επανδρωμένο Αεροχήμα (UAV), Χαρτογράφηση Βλάστησης, Παρακολούθηση Στόχων, Επεξεργασία Γεωχωρικών Δεδομένων, PMV-Geo, Σημασιολογική Κατάτμηση, Παρακολούθηση Πολλαπλών Αντικειμένων, Δορυφορικές Εικόνες

Contents

1 Introduction **11**
- 1.1 Geolocation Estimation and Constraints of Geospatial Mapping . . . 12
- 1.2 Contribution . . . 14
- 1.3 Thesis Outline . . . 15

2 Technical Foundations of Computer Vision **16**
- 2.1 The Vanishing Gradient Problem . . . 17
- 2.2 The Black Box Problem . . . 18
  - 2.2.1 Explainability & Perception . . . 18
- 2.3 Convolutional Neural Networks (CNN) . . . 19
- 2.4 Vision Transformers (ViT) . . . 21
- 2.5 Comparative Performance of CNNs & ViTs . . . 22
- 2.6 Architectural Evolution and Structural Innovations of YOLO Models . . . 24
  - 2.6.1 First Generation Modern Architectures: YOLOv4 and Scaled-YOLOv4 . . . 24
  - 2.6.2 Architectural Redefinition and Re-Parameterization: YOLOv7 . . . 26
  - 2.6.3 Anchor-Free Detection and Gradient Optimizations: YOLOv8 . . . 27
  - 2.6.4 Breaking the Information Bottleneck: YOLOv9 . . . 28
  - 2.6.5 Shift to NMS-Free E2E Object Detection: YOLOv10 . . . 29
  - 2.6.6 Complete Architectural Re-definition: YOLOv11 . . . 30
  - 2.6.7 Attention-Centric Spatial Modeling: YOLOv12 . . . 30
  - 2.6.8 YOLO26: Edge-First Small-Target Optimized NMS-Free DFL-Free Architecture with MuSGD Training . . . 31
- 2.7 Synthesis of YOLO Architectural Advancements . . . 33

3 Computer Vision in Aerial Imagery **34**
- 3.1 Challenges of Computer Vision in Aerial Imagery . . . 34
- 3.2 Public Domain CV Datasets for Aerial Imagery . . . 36
- 3.3 Comprehensive Analysis of the VisDrone Dataset . . . 38
- 3.4 Computer Vision Performance Metrics . . . 39
- 3.5 YOLO Model Performance on VisDrone dataset . . . 40
- 3.6 Architectural Improvements for Small Target Detection . . . 41
  - 3.6.1 Global Context through Area Attention and Partial Self Attention . . . 42
  - 3.6.2 Multi-Scale Adaptation . . . 42

4 Geolocation Estimation Pipeline **44**
- 4.1 Study UAV and Intrinsic/Extrinsic Parameters . . . 44
- 4.2 Asymmetric Sensor Cropping and Principal Point Displacement . . . 46
- 4.3 Pipeline Implementation and Lens Distortion Modeling . . . 47
  - 4.3.1 Extrinsic Parameter and Hardware Constraints . . . 47
- 4.4 Projection of 2D Image Plane to Global Coordinate System . . . 49
  - 4.4.1 Established Image Principal Point Coordinates . . . 50
  - 4.4.2 Target Pixel Displacement from Principal Point . . . 50
  - 4.4.3 Gimbal Rotational Transformation . . . 51
  - 4.4.4 Ray To Ground Intersection . . . 53
  - 4.4.5 Solving the Depth Problem . . . 53

4.4.6 Scaling Meters to Geodetic Degrees . . . 53

5 Experimental Geolocation Applications and Advanced Mapping Techniques 55
5.1 Application I: Probabilistic Multi-View Geolocation of Semantically Segmented Salt Marshes . . . 55
5.1.1 Ecological Importance of Salt Marshes . . . 56
5.1.2 Monitoring of Coastal Salt Marshes . . . 57
5.1.3 Monitoring Coastal Marshes through UAV Imagery . . . 57
5.1.4 Data Acquisition . . . 58
5.1.5 Computer Vision Pipeline Training & Evaluation Dataset . . . 59
5.1.6 Geolocation Pipeline Evaluation Dataset . . . 60
5.1.7 Detection of Coastal Salt Marshes . . . 61
5.1.8 Probabilistic Multi-View Geolocation using Gaussian-Weighted Rasterization . . . 65
5.1.9 PMV-Geo Evaluation Framework . . . 68
5.1.10 Symmetric Cross-Correlation Evaluation (SCCE) . . . 69
5.1.11 Structural Symmetry Scoring algorithms . . . 69
5.1.12 Isolating Geolocation Error to Evaluate Symmetry Scoring Techniques . 71
5.1.13 Results . . . 71
5.1.14 Application I: Conclusion . . . 75
5.2 Application 2: Geolocation Estimation of Detected and Tracked Targets in Aerial Footage . . . 77
5.2.1 Core Contributions . . . 77
5.2.2 Geolocation Evaluation Framework . . . 78
5.2.3 Data Acquisition & Geolocation Evaluation Scenarios . . . 79
5.2.4 Training and Detection of Optimized Networks . . . 80
5.2.5 Detection and Geolocation of Vehicles and Pedestrians . . . 80
5.2.6 Geolocation Evaluation Results . . . 82
5.2.7 Application II: Conclusion . . . 86
5.3 Concludion & Discussion . . . 86

Chapter 1: Introduction

Unmanned Aerial Vehicles (UAVs) have emerged as essential platforms across environmental monitoring, search and rescue, precision agriculture, and security surveillance. Equipped with high-resolution RGB and thermal sensors, aerial platforms provide unmatched operational flexibility and spatial resolution compared to satellite remote sensing or manual ground surveys. However, converting raw visual streams into actionable spatial intelligence requires closing the gap between image-plane perception and global geographical mapping. Modern deep learning architectures specifically Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) have achieved state-of-the-art performance in identifying, tracking, and segmenting targets within 2D image coordinates. Transforming these localized 2D predictions into precise geodetic global coordinates (such as WGS-84) introduces complex geometric and photogrammetric challenges. This thesis explores these concepts and introduces two real-world applications, the first adopts semantic segmentation to detect vegetation types in aerial imagery and through geolocation map them. The second application adopts object detectors and tracking models to detect objects (e.g. vehicles) for aerial footage and projects them into world maps. Figure 1, illustrates the design of both applications.

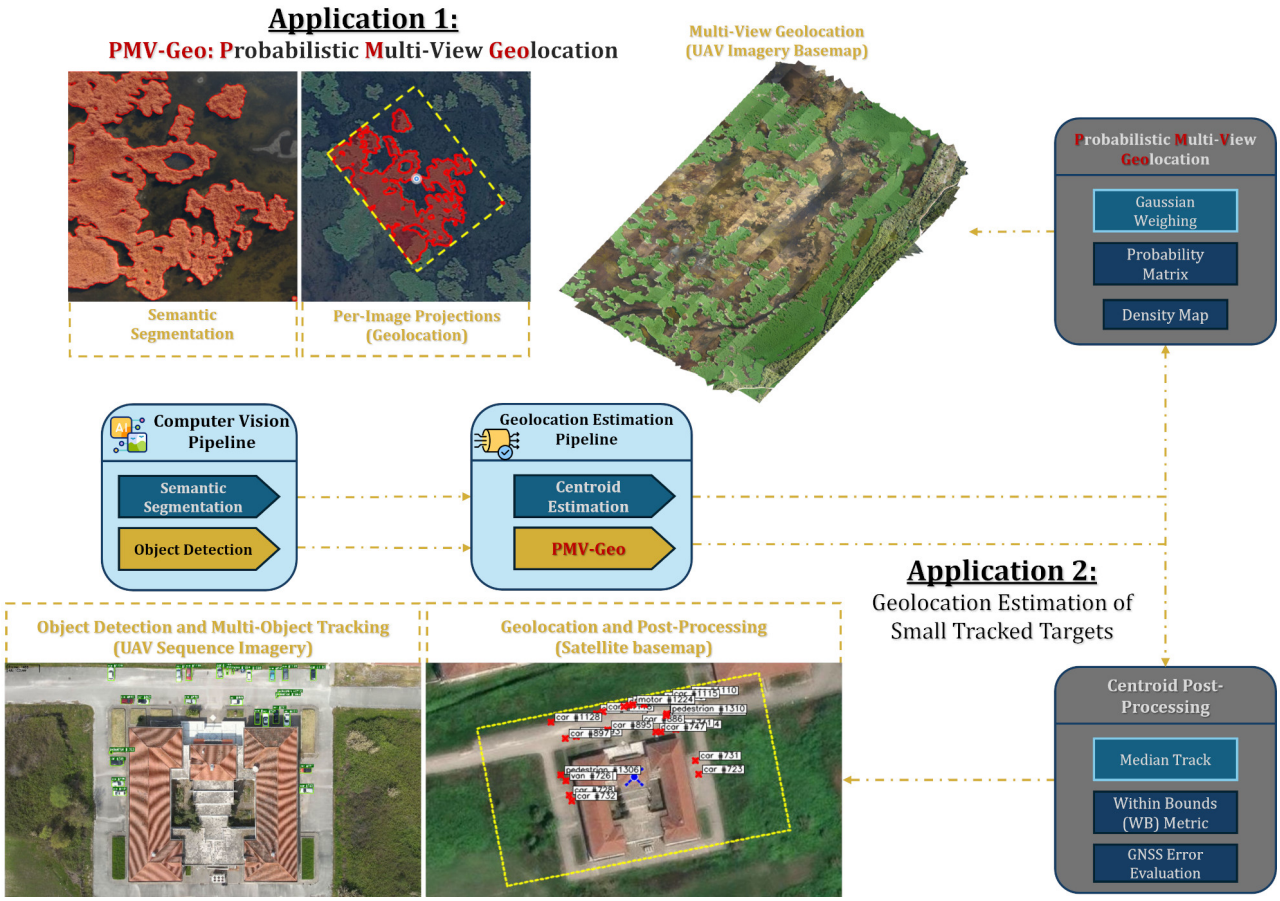


Figure 1: Workflow Design of Two Practical Applications Integrating Computer Vision and Geolocation Estimation

1.1 Geolocation Estimation and Constraints of Geospatial Mapping

Geolocation estimation in monocular UAV photogrammetry defines the process of transforming a 2D pixel coordinate (u, v) from an image frame into a 3D geographical coordinate vector (ϕ, λ) on the Earth's surface. Unlike stereoscopic vision or multi-sensor LiDAR setups, monocular camera pipelines lack direct 3D depth measurements. Consequently, optical ray projection relies strictly on onboard UAV telemetry (GNSS positioning, relative altitude, and gimbal pitch/yaw/roll angles) alongside camera intrinsic parameters. In operational real-world deployments, accurate monocular geolocation estimation faces several critical physical and structural constraints:

- **Target Tracking through Implanted GPS devices**

  These applications are often challenging needing to plant GPS devices into objects (e.g. animals, wildlife, aquatic life). While this method is often reliable in certain applications, GPS precision is often limited to 5-10 meters for such small devices and can often be costly and time consuming.

- **Mapping Geospatial Data through Satellite Imagery**

  Satellite imagery often contain geospatial data allowing for monitoring of objects. However, most commercially available satellite images provide resolutions up to 10 m^2 per pixel such as Seninel-2, thus, making monitoring of small objects challenging. Other alternative ultra high resolution satellite imagery including WorldView Imagery (2 and 3), provide ultra high resolution of up to 0.3 m^2 per pixel, however, they are often extremely expensive for researchers to acquire.

- **Lack of Depth in RGB Imagery**

  The human body and several other organisms often have two eyes to perceive depth, this phenomenon is addressed as Binocular vision [1]. Depth enables the distance estimation of an object from the lenses. Conventional monitoring methods leverage monocular cameras to map objects, however, RGB cameras are unable to perceive depth and distance from lenses to objects. LIDAR sensors can distinguish depth, however, they often lack the ability to recognize RGB features that are crucial for several object types (e.g. vegetation, vehicle color, etc). Moreover, phenomenons such as perspective distortion [2, 3] will aggravate GPS drifting based on the elevation of an object, depth can effectively minimize. For monocular cameras, several methodologies can be adopted to recognize depth. Either through post-processing using Gaussian Splatting [4] or to integrate Digital Elevation or Depth Estimation models.

- **Lens Distortion**

  Monocular camera lenses are often structurally designed with a pincushion or barrel effect to concentrate light rays directly into the sensor, thus, enabling digital cameras to produce an image. This curvature results to the image appearing slightly distorted based on the lens design. In geolocation estimation, this pincushion/barrel effect, can result in slight GPS displacement error based on the effect. Figure 2, illustrates the lens distortion effect, thus, resulting in the image appearing distorted. This effect can be corrected using radial and tangential distortion coefficients that are usually provided by

the camera manufacturer. If not provided, a user is required to correct for it using several methodologies such as with a checkerboard pattern [5].



Figure 2: Lens Distortion Effect and correction

- **LIDAR Sensors in UAVs**

  Several monitoring methodologies for vegetation types using UAVs are conducted with high resolution Light Detection and Ranging (LIDAR) sensors. LIDAR sensors synthesize Point Clouds by measuring the time of distance of a ray's reflectance from an object to the sensor. Using the Point Cloud, a 3D space is designed and leveraged to geospatially map out objects and vegetation. One study [6] conducted Wetland monitoring surveys by fusing Multispectral and Lidar sensors to classify Salt Marsh Vegetation. Such commercial drones equipped with high resolution LIDAR sensors are extremely expensive but often achieve really good accuracy for detection. Figure 3, illustrates the fusion of LIDAR and RGB camera sensors to effectively map salt marshes.

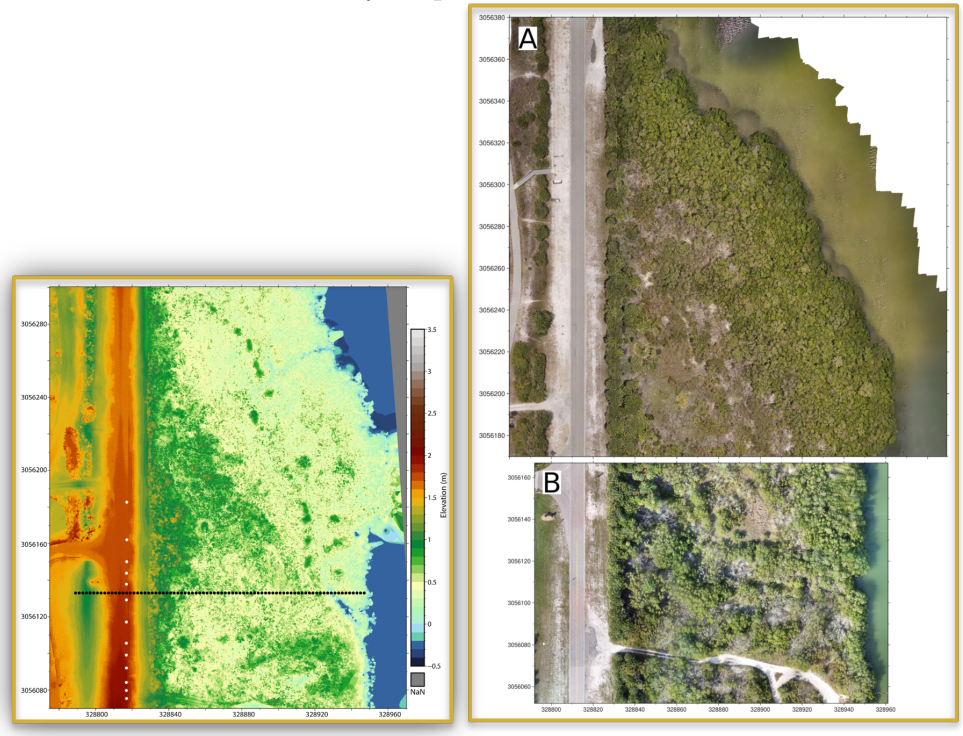


Figure 3: Digital Projections of Terrain Elevation using LIDAR and Orthomosaic Visualization

This thesis addresses several of these challenges by combining advanced deep learning target detection models with geolocation estimation methodologies.

1.2 Contribution

This thesis presents a unified, comprehensive framework that integrates modern edge-optimized computer vision architectures with rigorous photogrammetric projection pipelines for automated UAV-based geolocation. The primary contributions of this work are summarized as follows:

1. **Utilization of Unmanned Aerial Vehicles**

   This study leverages the commercial Drone DJI Matrice 30 Thermal, that is equipped with three types of camera sensors (Wide, Zoom, Thermal). The study goal is to utilize the Wide RGB camera sensor for the detection of objects and their subsequent projection into geospatial coordinate maps.

2. **Merging Visual Perception from Drones with the 3D Space**

   Development of a complete mathematical pipeline mapping 2D image pixels to WGS-84 global coordinate systems, explicitly modeling boundaries of objects visible within the camera sensor into world maps by leveraging parameters such 3D gimbal orientation matrices, barometric altimeter metrics, GNSS Positioning and lens distortion coefficients.

3. **Fusion of Computer Vision and Geolocation Estimation**

   Automating the task of detecting object boundaries using advanced edge-optimized high precision deep learning models. This study adopts modern state-of-the-art edge-optimized YOLO (You Only Look Once) models [7] for high precision and performance inference and is combined using the mathematical pipeline of Geolocation Estimation for drone imagery.

4. **End-to-End Geolocation Estimation Applications**

   This thesis introduces two practical end-to-end frameworks fusing Computer Vision and Geolocation, along with advanced post-processing methodologies for target detection and tracking in UAV footage and advanced vegetation mapping through aerial imagery.

1.3 Thesis Outline

The remainder of this thesis is structured into the following chapters:

- **Chapter 2 — Technical Foundations of Computer Vision:** Reviews core deep learning principles, including the vanishing gradient problem, model explainability (Grad-CAM), CNNs, Vision Transformers, and a systematic architectural review of YOLO models from YOLOv4 through YOLO26.

- **Chapter 3 — Computer Vision in Aerial Imagery:** Details domain-specific challenges in drone imagery (scale variations, perspective distortion, weather shifts, and receptive field attention), evaluates public UAV benchmarks (VisDrone, UAVid, FloodNet, UAVDT), and presents architectural optimizations for small target detection.

- **Chapter 4 — Geolocation Estimation Pipeline:** Provides the mathematical and physical derivation of the proposed 2D-to-3D projection pipeline, detailing intrinsic/extrinsic parameter modeling, asymmetric crop adjustments, Brown-Conrady distortion modeling, gimbal rotation matrices, ray-ground intersections, and WGS-84 geodetic conversion.
- **Chapter 5 — Experimental Geolocation Applications and Advanced Mapping Techniques:** Presents two real-world operational applications: PMV-Geo for fine-grained salt marsh canopy mapping evaluated via a custom algorithm entitled SCCE, and real-time aerial vehicle and pedestrian tracking/geolocation evaluated under controlled staircase and non-static flight profiles. The chapter concludes with a synthesis of findings and directions for future research.

Chapter 2: Technical Foundations of Computer Vision

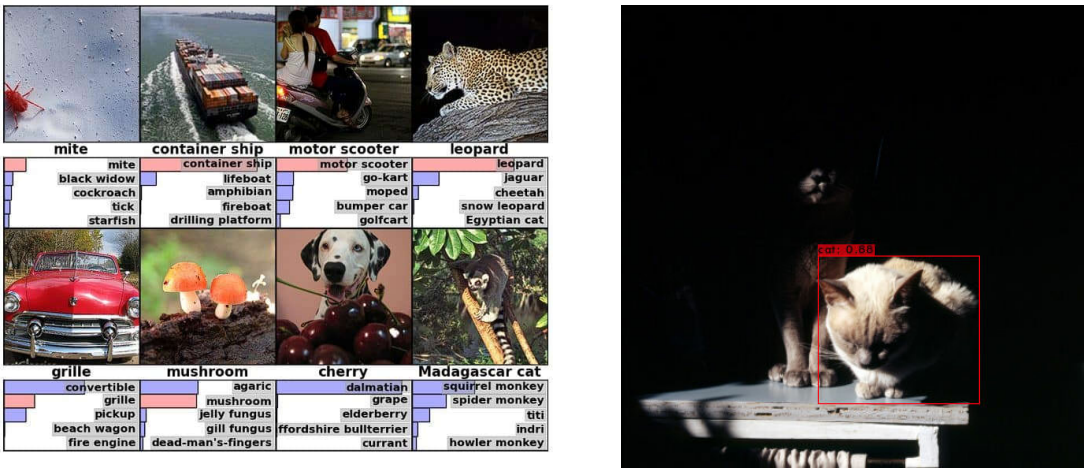
The task of estimating the geographic coordinates of a target and automating in UAV Photogrammetry requires the combination of two domains: Computer Vision and Geometry Projection. While Computer Vision effectively solves the **what** and **where** in an image, Geometry Projection also referred as Geolocation Estimation is new domain being explored in this thesis. Computer vision is an extensive field based on Artificial Intelligence used to automate image recognition and detection of patterns in images. It consists of various applications with each one employing different methodologies to extract patterns from images or videos. The most prominent methods of computer vision is with the use of Convolutional Neural Networks (CNNs). CNNs are trained on large datasets to perform tasks like:

- **Image Classification:**

  The process of classifying a whole image as a single class or label. A well-known study in 2014, revolutionized computer vision by proposing a large classification dataset called ImageNet [8]. The dataset consists of 15 million images and approximately 22,000 categories. Figure4a is an image classification example of various objects.

- **Object Detection:**

  The process of detecting the location of multiple objects in an image. It is often used to separate two or more types of objects (e.g. person, vehicle, bicycle). In practice, the models will produce Bounding Boxes describing the location, width and height of an object in an image. One study [9] utilized a standard computer vision model namely YOLOv4 [10] to detect objects in low light images while simultaneously evaluating image enhancement techniques (Figure 4b).



(a) Image Classification src: [8]

(b) Object Detection src: [9]

Figure 4: Traditional Computer Vision Applications: Image Classification and Object Detection

- **Semantic Segmentation:** Semantic Segmentation (SS) is the task of producing pixel-level classification masks of pixels corresponding to specific classes within images. SS is often employed in advanced applications such as Medical Imaging, Satellite Imagery and UAV imagery applications. One such application includes a study [] that was conducted

to detect Salt Marshes in Satellite Imagery using the DeepResidual-CBAM model as illustrated in Figure 5.

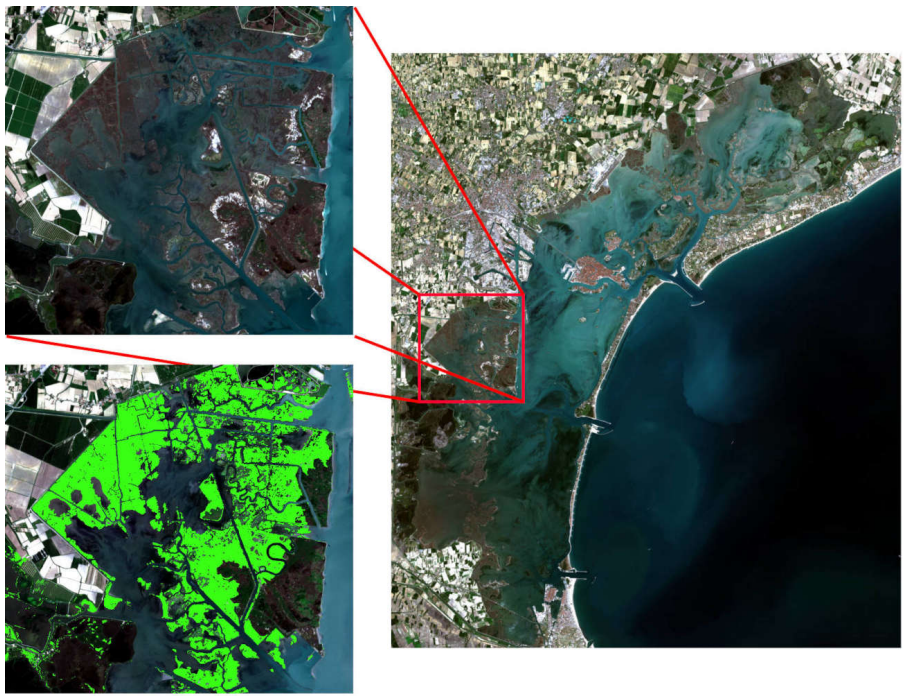


Figure 5: Detection of Salt Marshes using a Attention-Enhanced Residual Dense Network

Within the last decade (2015-2025) the field of Computer Vision is characterized by a relentless pursuit of architectural optimizations to improve the accuracy of these tasks. A considerable amount of research is designed for the development of Deep Neural Networks knowing that more depth resulted to greater interpretation and feature extraction capabilities. Those networks were often trained using backpropagation by dynamically adjusting the weights of the model. However, training these models became significantly more challenging due to the vanishing gradient problem.

2.1 The Vanishing Gradient Problem

Since the year 1998, this problem was widely acknowledged by a number of researchers as formally described in a study for recurrent-neural networks and its implications [11]. The study theoretically analyzes how the gradient slowly decays when propagated backwards into the model through many recurrent steps. Meaning, the deeper the network, the more shrinkage of the gradients, resulting in sub-optimal layer updates, making training all the more difficult. In 2015, the field was expanding on new architectures while recognizing the limitations of increasing depth. Although deeper networks theoretically possessed richer feature interpretation, training them was impossible due to the vanishing gradient problem. The ResNet model [12] was the first CNN model to solve the Vanishing Gradient problem, by including a new revolutionary training mechanism namely Residual Learning. Its major addition were shortcut connections from the network input to the output. This new breakthrough sparked an explosion of deeper neural network architectures particularly for Computer Vision. Various other modern technologies address this problem through architectural solutions often found in modern CNNs

like DenseNet [13] and EfficientNetv2 [14], Vision Transformers like DeTr (Detection Transformer) [15] and R-CNNs like Faster-R-CNN [16] (Faster Region-based Convolutional Neural Network).

2.2 The Black Box Problem

The modern development of Artificial Neural Networks (ANNs) is largely characterized by a competitive pursuit of state-of-the-art performance. This methodological drive, adopted by both industry and academia, favors the production of increasingly larger and structurally more complex models only to get the role of the best performing model. The resulting architectures, often defined by immense scale, billions of parameters, and highly non-linear transformations, represent a significant engineering achievement. While this approach of escalating complexity yields unprecedented gains in accuracy, it introduces a profound loss of interpretability. This phenomenon, namely the "Black-Box Problem", refers to the difficulty in understanding how complex algorithms, particularly ANNs, arrive at their decisions. This opacity fundamentally limits credibility and hinders vital model refinements that would otherwise be guided by clear model explainability and perception. To counter the black-box phenomenon, many researchers have shifted their attention towards approaches that provide better explainability towards how the model came to its conclusion.

2.2.1 Explainability & Perception

Following the challenges associated with the "Black Box Problem", the sub-field of Explainable Artificial Intelligence (XAI) provides essential methodologies for interpreting complex model behaviors. In the domain of Computer Vision, deep learning architectures such as Convolutional Neural Networks (CNNs) [7] and Vision Transformers (ViTs) [17] have established themselves as the primary tools for extracting semantic information from visual data. While these models achieve state-of-the-art performance in standard tasks like object detection and image classification, their internal decision-making processes remain historically opaque. This lack of transparency poses a significant challenge in high-stakes applications where understanding the rationale behind a prediction is crucial for scientific and safety validation. A prominent technique to interpret model perception in the Computer Vision Domain is through the use of Gradient-weighted Class Activation Mapping (Grad-CAM) [18]. Grad-CAM generates heatmaps that visualize which spatial regions of an input image were most influential in the model's final prediction. By applying this technique across the network layers, researchers can observe how the model transitions from detecting low-level visual features, such as edges and textures, to high-level semantic objects. This transparency is essential to validate and refine model performance, ensuring the reliability and interpretability of automated decision-making systems.

2.3 Convolutional Neural Networks (CNN)

During this explosion of new architectural designs in Computer Vision, the first major architectures that were introduced is the YOLO model (You Only Look Once) [7], DenseNet [13], Inception-v3/v4 [19] [20] and MobileNet [21] in the years 2016-2017. These models were designed specifically for applications such as Object Detection and Image Recognition. Figure 6, illustrates the standard YOLO [7] architecture based on the GoogLeNet [22] backbone.

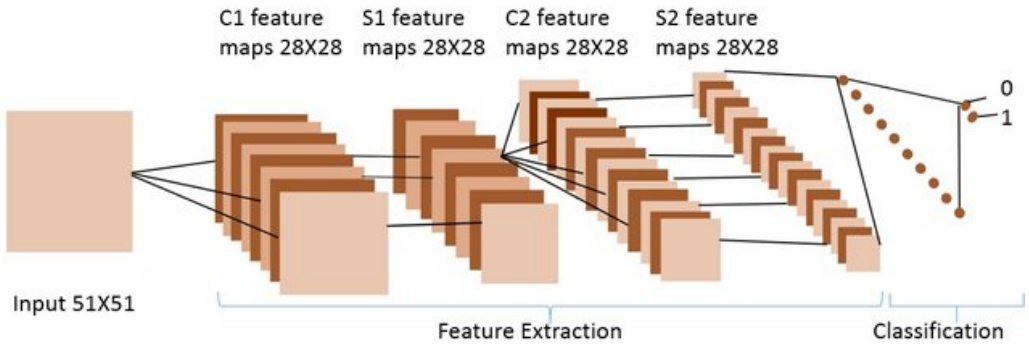


Figure 6: YOLO CNN Architecture

The standard CNN architecture consists of three main layers: Input, Feature Extraction and Classification.

Input Layer:

The input stage of a CNN is fundamentally defined by the geometric shape of the input vector that is fed into the network. This vector typically consists of a 2 or 3 dimensional structure with various resolutions (e.g. 640×640, 1920×1080, 4096×2160). In some scenarios where the input vector does not align with the constraints of the network, the input is often resized to match the network's size. The shape of the vector is typically the same in training and inference to not influence the accuracy of the model. However, while it is often suggested to use matching vector and network shapes, it often varies on the application and goal. Assuming the goal is to detect small objects, by utilizing matching shapes, the network will utilize the full quality of the vector without impacting performance. However, if the goal is to detect large objects, it is often suggested to use smaller network input in comparison to the vector. This will indirectly benefit the model's output by reducing noise or artifacts within the vector. This is backed by various studies proving how increasing/decreasing network input size inversely impacts the performance of large/small objects in images [23], [24], [25]. Although the two type of applications produce inverse results, more investigation is required to produce a model that produces desirable results for small and large objects respectively.

Feature Extraction Layer:

The Feature Extraction stage, aka the hidden layer, operates by alternating between Convolutions [26], Pooling layers [27], Batch Normalization [28] and Activation Functions (ReLU) [29]. This layer, located between the input and output layers performs dimensionality reduction by extracting the most prominent features through convolutions and finally leading to the model's prediction.

1. **Convolution:** The core mechanism of CNNs are the Convolutions hence the name. They operate by multiplying input pixels with a small matrix namely Kernel. Various kernel sizes may vary from 3×3, 5×5 and 7×7. The kernel works by iteratively passing over each pixel through the image a process called "sliding". For each step, the kernel performs a dot product calculation with the pixel values of the image, effectively filtering the image to highlight essential patterns like edges, shapes and textures similar to contours or edge detection. The parameters of a standard Kernel filter is the Sobel Operator [30] to detect

vertical Edges:

$$K = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Typically, CNNs do not use a fixed Kernel filter, but instead learn them using backpropagation during training by adjusting them dynamically. The produced result once each pixel has been convoluted is the feature map.

2. **Pooling:** The Pooling Layer is employed to downsample the feature map by performing dimensionality reduction. Using the same sliding window technique, the layer aggregates the results onto a downsized feature map. *Maximum Pooling* is the standard approach, as it retains the most dominant features in the window. *Average Pooling*, is often used to reduce the spatial dimension of feature maps without significantly impacting the performance of the model.

3. **Batch Normalization:** Batch Normalization (BatchNorm) [28] is designed to solve the problem where each layer's input changes during training thus complicating the training process, this phenomenon also known as **internal covariate shift**. Batch Normalization improves training by adjusting the inputs of the previous layer into a consistent distribution, due to the "internal covariate shifting" nature of training a CNN.

4. **Activation Functions**: The Activation Function is applied at the end of a hidden layer to map the outputs of a feature map. It enables the model to learn complex patterns rather than simple linear regressions introducing non-linearity into the CNN. Figure 7 illustrates four activation functions and their corresponding equation.

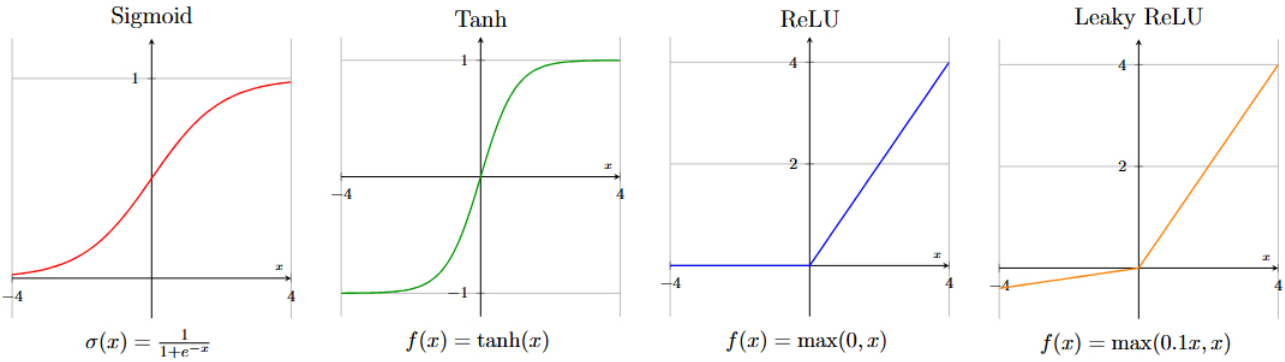


Figure 7: Comparison of Activation Functions.

While traditional activation functions like the Sigmoid and Hyperbolic Tangent (tanh) are often used in Neural Networks, they often suffer from the vanishing gradient problem. To address this, the Rectified Linear Unit (ReLU) [29] was introduced. An additional activation function was developed namely Leaky ReLU [31] that linearly scales any negative inputs, allowing for a more robust gradient optimization during ANN training. One study [32] conducted an experiment to evaluate and compare the performance of these Activation Functions on Poisson Noisy Image Classification data. The experiment was conducted on a number of CNN architectures, their findings reported that either ReLU and L-ReLU achieved the best performance depending on the CNN architecture.

Classification Layer:

The classification layer of a CNN is typically implemented using a fully Connected Network (FCN). Essentially, FCNs are Multi-Layered Perceptrons (MLP) [33] used to map low-level features extracted from the previous layer into high-level classes. The classification layer of a CNN is typically implemented using a Fully Connected Network (FCN), essentially acting as a Multi-Layered Perceptron (MLP) [33]. The FCN aggregates high-level features extracted from the previous layer and maps them into vectors of a specific format. For instance, the feature extraction layer will detect the edges of a vehicle in an image and further refine it, the FCN layer, will map these high-level feature maps into a fixed-length vector specific to the computer vision task or a single digit output classification-based tasks.

2.4 Vision Transformers (ViT)

With the introduction of word embeddings [35] and advances in Natural Language Processing, Transformers [34] were developed in 2017, leveraging self-attention mechanisms. Several years later, embeddings were adapted for images, enabling the extraction of information from them, these models titled ViTs (Vision Transformers) [36], were a major milestone, further expanding the capabilities of Computer Vision across the board. While traditional transformers processed as input 1D sequences of token embeddings, adapting transformers to process images was an occuring challenge. To combat this, reshaping the image into a sequence of flattened 2D patches was an effective solution to train transformers on images [36]. The architecture of these models integrates a encoder-decoder design [37] based on the output sequence and computer vision task. Figure 8, illustrates this design of a traditional transformer. The pre-processing step in vision-based transformers, works by processing the whole 2D image into 16×16 patches (smaller or higher) that are flattened into input sequences. Consequently, the input sequences are later linearly projected into patch embeddings to map raw pixels into a trainable embedding space. Prior to the encoder layer, positional embeddings are added to the patch embeddings to retain positional information as input to the encoder. Once fed into the encoder layer, it alternates through several self-attention layers to understand the relationship between the positional embeddings. These alternations enables the transformer to effectively extract a global "context" from these embeddings in a bi-directional method similar to BERT [38], where

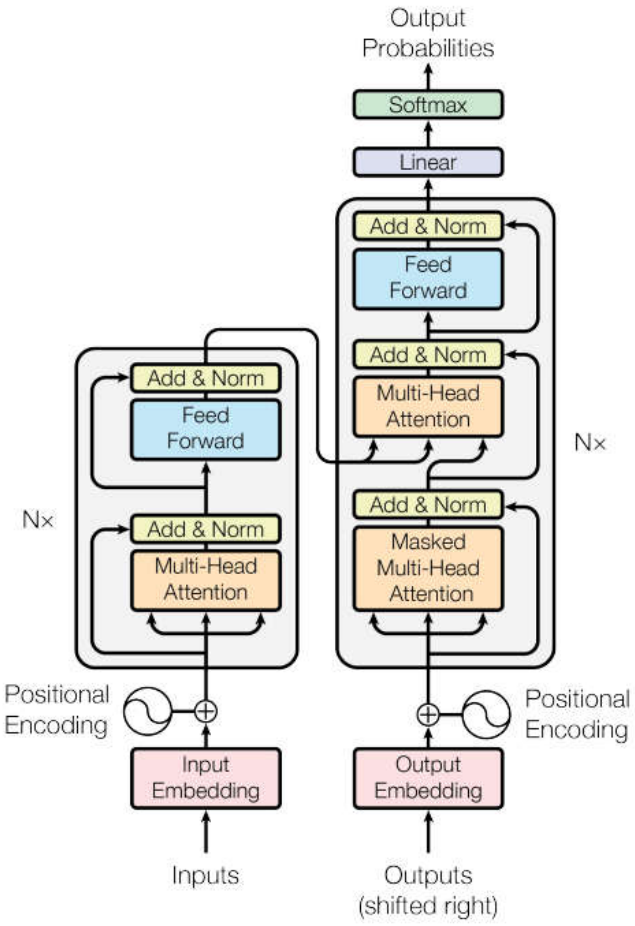


Figure 8: Encoder-Decoder Design [34]

each positional embedding contains information from adjacent embeddings. Like most CNN models, the final step of the encoder is to extract usable information from these positional embeddings, thus a MLP is employed at the end of the encoder to add non-linearity using the G-ELU activation function [39] that is often employed on Transformer architectures. Finally, decoders process the high level features extracted from the encoder by transforming, upsampling and reshaping them back into sequence formats for the appropriate task. The use of decoders for Vision Transformers in most CV tasks (Object Detection, Semantic Segmentation, etc), require a complex output structure, image classification requires a single output making the use of decoders redundant.

2.5 Comparative Performance of CNNs & ViTs

Convolutional Neural Networks (CNNs) and R-CNN-based models have long been the gold standard in extracting spatial information from images. In these architectures, local features are captured in early layers, while deeper layers gradually build a more global representation. This process introduces an inherent locality bias rather than broader context. Vision Transformers address this limitation by using self-attention mechanisms effectively capturing global relationships early and thus retrieving contextual information from the entire image. Several studies [15, 40, 41, 42, 43] report that ViTs generally outperform CNNs and R-CNNs in various computer vision tasks including. The following systematic review (Figure 1) demonstrates a comparative performance of state-of-the-art CNNs, R-CNNs and ViTs. Faster-RCNN, utilizes Region-Based Neural Networks with effective attention mechanisms further improving accuracy, however, these models are often more demanding in terms of computational performance. YOLO models compensate for the lack of performance and are built to perform real-time object detection tasks. Detr, ViT and Co-Detr, are all transformer-based models utilizing Encoder/decoder architectures.

Model	**Type**	**Backbone**	**params**	**mAP$_{\text{test}}$**	**mAP$_{\text{val}}$**
Faster-RCNN [16]	RCNN	VGG-16 [44]	2.4M	21.2	21.5
Resnet-101 [12]	CNN	Resnet	44.5M	-	27.2
YOLOv4 [10]	CNN	CSPDarknet53 [45]	27.6M	43.5	-
YOLOv7 [46]	CNN	E-ELAN [47]	36.9M	51.2	51.4
YOLOv8-m [48]	CNN	CSPDarknet53	25.9M	-	50.2
YOLOv12 [49]	CNN	R-ELAN [49]	20.2M	-	52.5
Detr [15]	Transformers	Resnet 50/101 [12]	60M	44.9	-
ViT-L [36]	Transformers	Resnet	361M	-	49.2
Co-Detr [50]	Transformers	Swin-L [37]	218M	-	60.7
Co-Detr	Transformers	ViT-L [36]	304M	66.0	65.9

Table 1: Systematic Comparative Performance Review of state-of-the-art CNNs, R-CNNs and ViTs on the COCO dataset.

While vision-based transformers provide higher performance than CNN and R-CNNs, there are several limitations and advantages that need to be addressed:

1. **Performance on small and large datasets:**

   Current literature reports that vision Transformers generally require larger datasets to effectively train them compared to CNNs [41, 51]. While CNNs tend to generalize better with limited data, studies show that ViTs benefit more from pre-training, fine-tuning and augmentation, which can significantly improve their performance.

2. **Multi-Modal Flexibility**:

   Unlike Convolutional Neural Networks which are architecturally limited to process primarily images, Vision-based transformers treat inputs as sequences of embeddings. This enables for the integration of diverse data types such as text, audio and images into a unified transformer model by leveraging an encoder-decoder design. This flexibility has led to the development of Multi-Modality Learning and tasks such as Visual Question Answering (VQA) [52], where a transformer model synthesizes information from natural language and images. ViTs are also used in broader image processing tasks such as image generation through stable diffusion [53], Medical Imaging using semantic segmentation to detect diseases [42]

3. **Training time to Convergence**:

   Convergence refers to the training state where a model's performance stabilizes and further training results in minimal to no improvement. At this stage, the optimization process has plateaued, indicating that the model has reached its maximum learning potential for the given data and training configuration. Convergence deviates across CNNs and ViTs. One study

4. **Explainability & Perception**:

   Explainability and model Perception is an important concern among the growing architectures of Artificial Intelligence models. CNNs often operate as black boxes, meaning the provide limited information on how the CNN's results were concluded and are difficult to interpret them. However, Vision-based transformers

2.6 Architectural Evolution and Structural Innovations of YOLO Models

The YOLO object detection models were selected for this task due to their efficiency in real-time inference and their continuous architectural improvements to improve computational performance and accuracy. YOLO models employ a fundamentally different approach compared to traditional two-stage detectors like Faster R-CNN [16]. YOLO formulates predictions by processing the entire image in a single forward pass through a unified CNN architecture. This approach enables significantly faster inference speeds while maintaining competitive performance. Recent major contributions in the YOLO architecture design includes the removal of the Non-Maximum Suppression (NMS) function [54]. NMS is traditionally employed in the last step of the inference to filter anchor boxes and utilize predictions containing the highest confidence value.

2.6.1 First Generation Modern Architectures: YOLOv4 and Scaled-YOLOv4

The introduction of YOLOv4 [10] marked a critical milestone in real-time object detection by systematically integrating a "**Bag of Freebies**" and a "**Bag of Specials**". Following these contributions, "Bag of Freebies" refers to a set of training strategies consisting of data augmentation (e.g. Mosaic and CutMix) and loss function adjustments (CIoU loss) by increasing training costs without impacting inference latency. The latter, "Bag of Specials", refers to an architectural modification by including architectural blocks (e.g. Cross-Stage Partial CSP, Spatial Pyramid Pooling - SPP, Self-Attention Modules - SAM, and Path-Aggregation Network - PAN) and post-processing methods (DIoU-NMS) that marginally increase inference computational cost but significantly improve precision. YOLOv4 is composed of several parts (Figure 9)and integrates **Cross-Stage Partial** (CSP) network methodologies, the backbone combines **VGG16** [44], **ResNet-50** [12], **SpineNet** [55], **EfficientNet** [14], **CSPResNetXt50** [56] and **CSPDarknet-53** [45]. The neck of YOLOv4 employs a **Path Aggregation Network** (PANet) combined with **Spatial Pyramid Pooling** (SPP). The head of YOLOv4 employs **Dense Prediction** for a one-stage detector and **Sparse Prediction** (Faster-RCNN [16] and **R-FCN** [57]). Finally, the authors of YOLOv4 incorporate **Cross mini-Batch Normalization** (CmBN) a critical adjustment for aerial imagery.

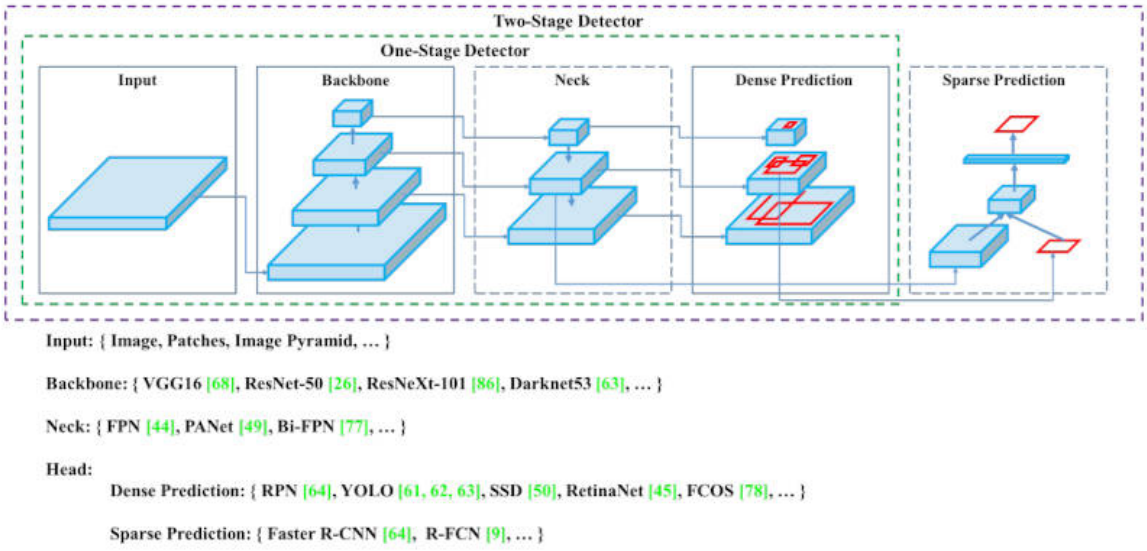


Figure 9: YOLOv4 Object Detector Architecture

Scaled-YOLOv4 [58] expanded upon the YOLOv4 architecture by transitioning from the Darknet [59, 60] framework to the Pytorch ecosystem [61] enabling more flexibility in architectural modification. The primary contributions of this model was the introduction of a systematic scaling method that uniformly adjusted the depth, width, and resolution of the network across the backbone, neck, and prediction heads simultaneously, hence the "Scaled" naming system. This scaling method is vital for aerial imagery, as it allows researchers to employ the exact same architecture on high-resolution UAV Imagery simply by adjusting scaling coefficients. To optimize for edge devices, Scaled-YOLOv4 introduced the **Cross-Stage Partial One-Shot Aggregation Network** (CSPOSANet) as backbone, specifically within the YOLOv4-tiny variant. This novel scaling methodology of adjustable coefficients was later adapted to modern YOLO architectures for greater customizability. Figure 10 illustrates the scalability variations

of YOLOv4.

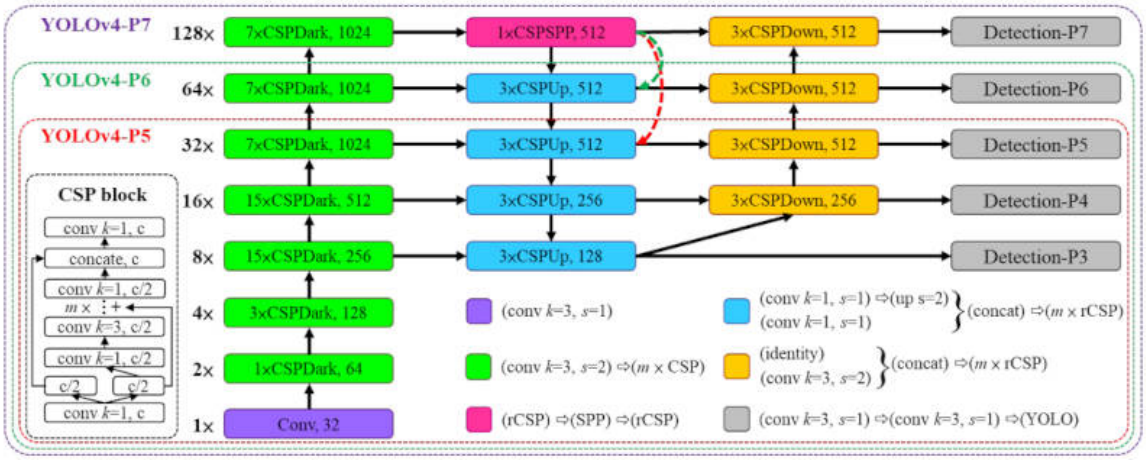
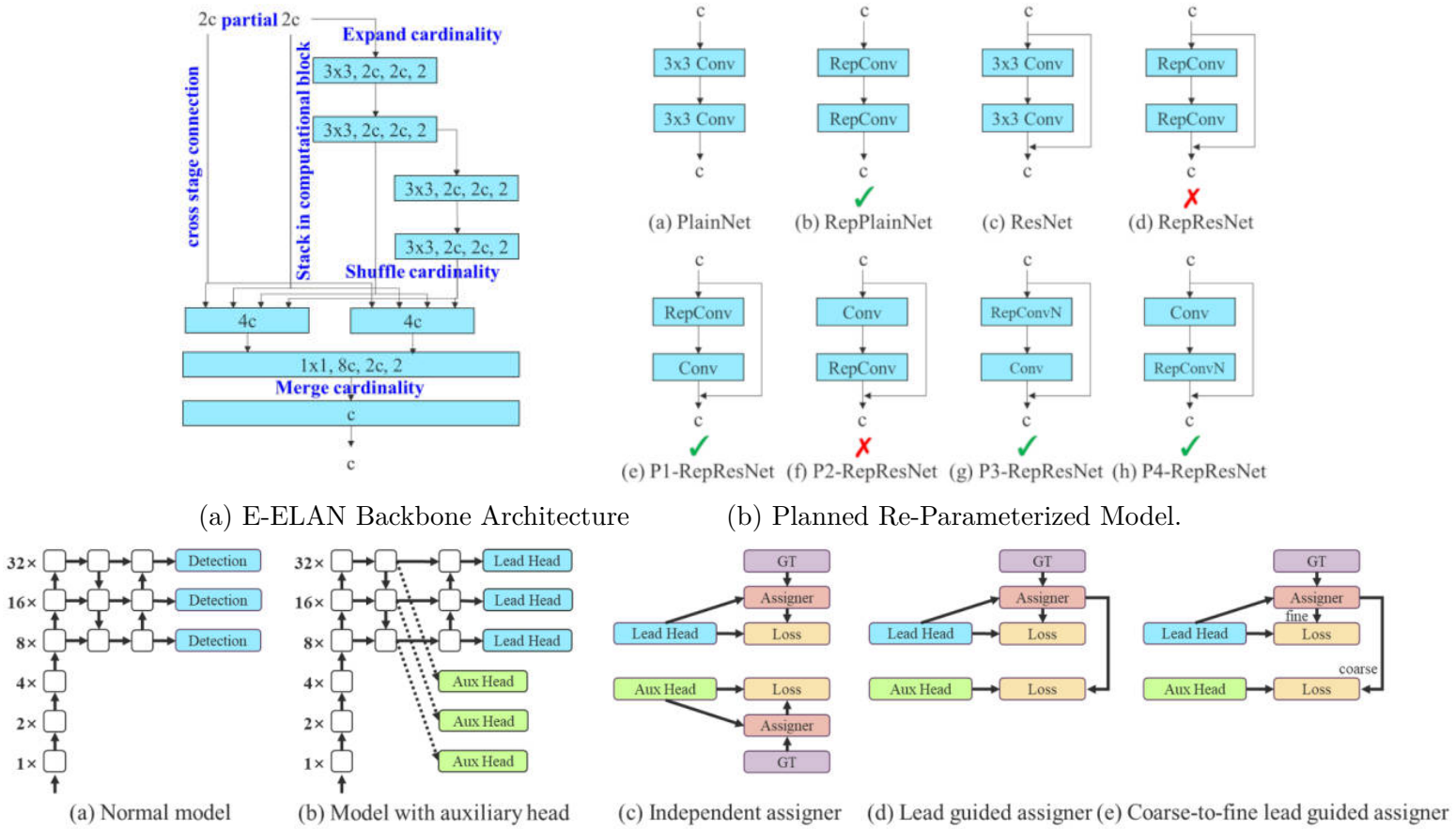


Figure 10: Scaled YOLOv4 Object Detector Architecture

2.6.2 Architectural Redefinition and Re-Parameterization: YOLOv7

YOLOv7 [46] redefined the standards for real-time object detection by shifting focus toward trainable network structures and gradient path optimization, achieving greater benchmarks. YOLOv7's core contribution was the transition from the traditional CSPDarknet53 backbone to the **Extended Efficient Layer Aggregation Network** (E-ELAN). E-ELAN improves gradient propagation pathing by employing shuffle and merge cardinality processes with different convolution blocks to learn features more diversely. YOLOv7 introduces **planned re-parameterized model** and a novel dynamic label assignment technology using **coarse-to-fine lead guided label assignment** mechanisms. Planned re-parameterizated convolutions allows the model to extract more refined and diverse gradients for different feature maps using the RepConvN, while the standard RepConv that combines 3×3 convolution, 1×1 convolution, and identity connections, it destroys residual connections in models like ResNet [12] and concatenation in DenseNet [13]. RepConvN removes the identity connections entirely. This design was derived specifically through gradient flow path propagation analysis, an impressive work developed by Chien-Yao Wang, and et. al. [62] that was later adopted to the newly founded planned re-parameterized model, YOLOv7. This novel design framework highlights the importance of model interpretability to refine model architectures, specifically through gradient flow propagation path analysis. Conversely, the coarse-to-fine lead guided label assignment mechanisms (Coarse for auxiliary and fine for lead loss) optimizes training for its multi-output architecture between its lead head (final output) and auxiliary heads (intermediate outputs). The study proposes a new label assignment method that guides both the auxiliary head and lead head to generate two distinct sets of soft labels based on the lead head's predictions and ground truth to create a hierarchical supervision structure. The E-ELAN novel backbone architecture is visually illustrated in Figure 11, along with the planned re-parameterized model and coarse-to-fine lead guided guided label assignment mechanism.



(a) E-ELAN Backbone Architecture (b) Planned Re-Parameterized Model.

(c) Coarse-to-fine label assignment mechanism map

Figure 11: Comprehensive structural visualization of YOLOv7 optimizations.

2.6.3 Anchor-Free Detection and Gradient Optimizations: YOLOv8

YOLOv8 developed by Ultralytics [63] is the first model to introduce anchor-free detection. Anchor boxes are pre-defined bounding boxes with specific widths and heights that server as reference for detecting objects of varying scales and aspect ratios. Instead of needing to predict absolute coordinates from scratch, the neural network adjusts these anchors, allowing the model to more accurately and efficiently detect objects by matching predicted boxes to ground-truth annotations using the Jaccard metric [64], also known as Intersection over Union (IoU). Additionally, YOLOv8 integrates a **Decoupled Head** and an advanced **C2F module**. Together, these modifications enable YOLOv8 to achieve superior performance and balance between accuracy and inference speed. Moreover, YOLOv8 returned back to the original **CSPDarknet backbone** [59] rather than using the novel backbone architecture [47], aiming for a more lightweight and structurally simple architecture.

Previous YOLO models leveraged architectures with coupled heads, meaning different tasks often conflicted with each other. This resulted in various predicting different tasks simultaneously, for instance classification scores (object classes) and regression scores (Bounding boxes), leading to sub-optimal training results, slower convergence and inference speeds. One study in 2021, developed the YOLOX model, the first object detector to introduce decoupled heads in the YOLO series, leading to state-of-the-art results across several YOLO models (Figure 12).

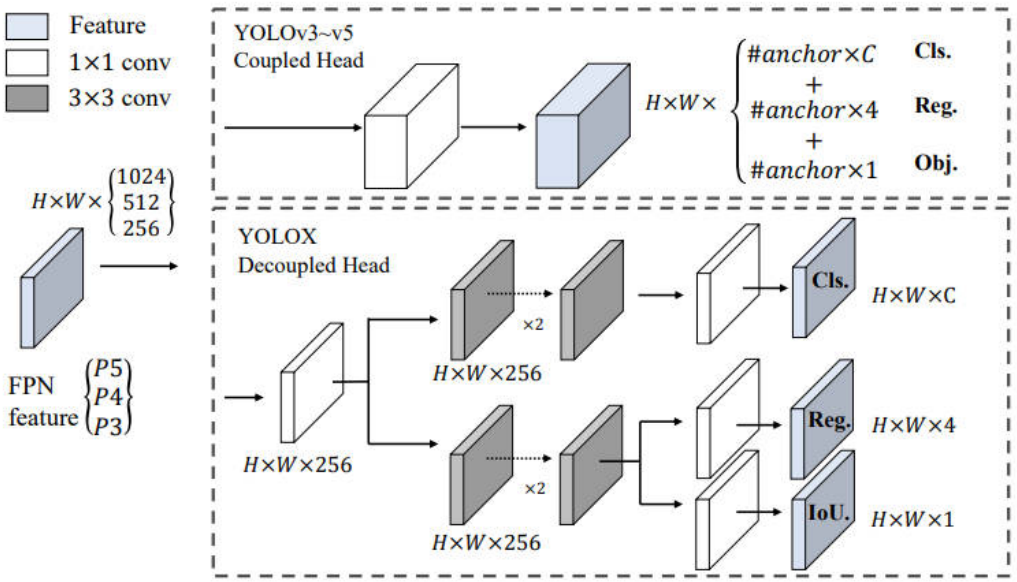


Figure 12: YOLOX Decoupled Head Object Detector

In 2023, YOLOv8 was released with similar features but greater performance, accuracy, customizability due to the Ultralytics framework and finally the addition of the advanced C2F module. The C2F module in YOLOv8 stands for Cross-Stage Partial with two convolutions (or Concatenate to Features) as the primary building block for the architecture's backbone. It replaces the C3 module used in YOLOv5 [63, 65] designed to optimize gradient propagation and feature diversity by retaining and concatenating intermediate outputs rather than discarding them. In practice, the C2F module leverages shortcuts (skip connections) to capture and split intermediate outputs from the internal bottleneck stack (3×3 convolution kernels) and concatenates them directly to the final convolution block output.

2.6.4 Breaking the Information Bottleneck: YOLOv9

As YOLO networks grew deeper in an attempt to capture greater semantic context, they increasingly fell victim to the Information Bottleneck Phenomenon [66]. According to the study, they monitored this phenomenon by plotting the Information Plane of Deep Neural Networks (DNNs) and observed that the network gradually forgot background details of the input data leading to information compression. The study debated that the "compression phase" is likely linked to DNN's success in generalizability and was likely caused by Stochastic Gradient Descent's (SGD) randomness [67]. Moreover, another study by Andrew Michael Saxe, et. al. (2018) [68], disputes that information bottleneck is not universal and that there is no evidence that information bottelneck improves generalization. Activation functions like ReLU completely mitigate this phenomenon, yet they are still capable of generalization, while other activation functions (tanh or sigmoid) exhibit compression due to the double-sided saturating nonlinearities generalize effectively too. Conversely, YOLOv9 [69] directly addressed this DNN phenomenon through the introduction of **Programmable Gradient Information** (PGI) and the **Generalized Efficient Layer Aggregation Network** (G-ELAN) as the backbone. These modifications resulted in higher accuracy with fewer parameters, while YOLOv8 offers superior inference speed due to leveraging a lighter backbone (CSPDarknet) in addition with the C2F module. PGI introduces a removable auxiliary reversible branch module into the network architecture that can be removed during inference while retaining the original network's capabilities.

This ensures that input information maps directly to the targets, allowing the model to obtain reliable gradient information from the auxiliary reversible branch. Consequently, the G-ELAN backbone is a combination of two neural network architectures CSPNet [56] and ELAN [70] that was designed as a lightweight and accurate model. This new architecture allows for the use of any computational block to be integrated along the network.

(a) Generalized Efficient Layer Aggregation Network (G-ELAN) Backbone Architecture

(b) Programmable Gradient Information (PGI) Module

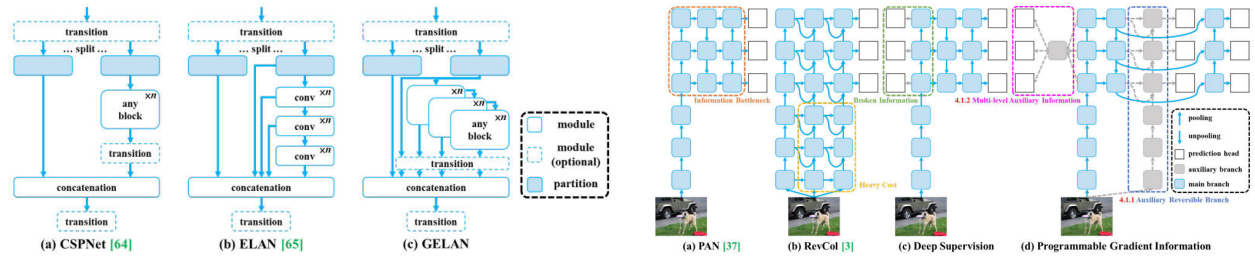


Figure 13: Structural Components and Optimizations of YOLOv9. (a)

2.6.5 Shift to NMS-Free E2E Object Detection: YOLOv10

YOLOv10 [71] is the first YOLO model to entirely remove the Non-Maximum Suppression (**NMS-Free**) post-processing function through **End-to-End** object detection achieving substantial inference speeds in comparison to previous YOLO architectures. This was achieved through the **Dual Label Assignment** method by leveraging both **one-to-many** and **one-to-one** predictions to ground-truth. Where one-to-one assignment uses a single prediction for each ground-truth, thus avoiding the NMS post-processing entirely. During training, the study incorporated two heads that jointly optimized the model using both assignments one-to-one and one-to-many for richer supervision. During inference, one of the heads was discarded (one-to-many) to utilize only a single head. This jointed training strategy (Figure 14) yielded significantly faster inference overall, eliminating the need to use NMS.

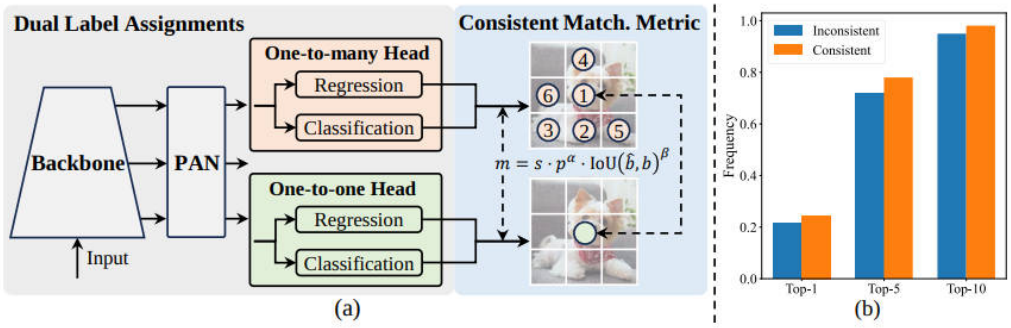


Figure 14: YOLOv10 Training Consistent Dual Assignment Strategy

Through my personal experience using YOLOv10 and due to its specific architectural trade-offs designed for inference speed, detections were highly inconsistent across many frames in video footage. While the model converged relatively quickly, it consistently failed to generalize effectively across several datasets resulting in said inconsistencies. Moreover, v10 fails at particularly complex tasks or small objects. While the concept of Consistent Dual Assignment is in-genius by enriching the features during training and converging significantly quicker

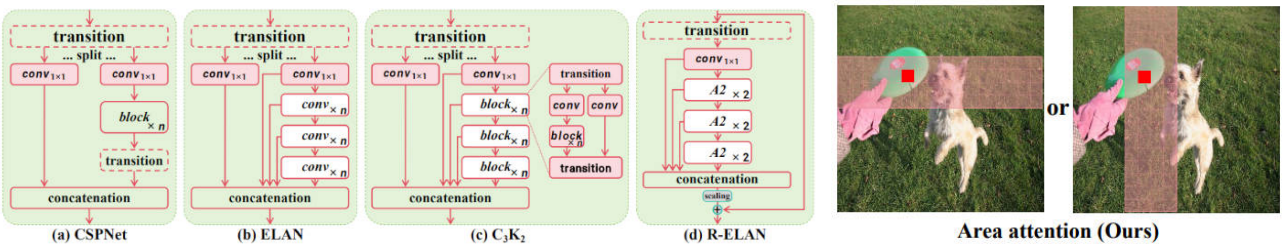
than other models, it fails to meet the standards of other state-of-the-art YOLO models. One study documented YOLOv10's inconsistiencies along several other YOLO models in Complex Orchard Environments [72].

2.6.6 Complete Architectural Re-definition: YOLOv11

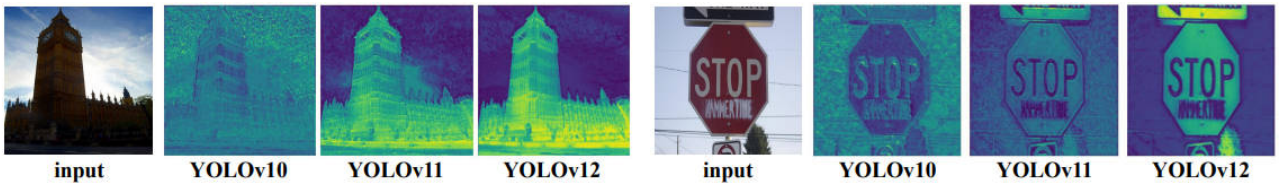
YOLOv11 [73] presents a direct upgrade to the YOLOv8 model by further improving model performance in several ways prioritizing on enhanced feature extraction in addition with spatial attention enhancement. This model introduces a new **Cross Stage Partial with Spatial Attention** (C2PSA), it replaces the C2F module of v8 with the **Cross Stage Partial with kernel size 2** (C3k2) block. Notably, while YOLOv11 offers E2E training capabilities, it is strictly by default not NMS-Free like v10. The C3k2 block is a computationally more efficient and compact implementation of the Cross-Stage Partial (CSP) bottleneck. It employs two smaller convolutions as dictated by the naming system (2×2 kernel size). Conversely, the C2PSA block is an addition enhancing model spatial attention within the feature maps, enabling the model to prioritize in critical regions of interest, thus potentially improving detection accuracy. Several studies have leveraged YOLOv11 for remote sensing and aerial imagery to optimize and improve the model architecturally and thus the task of small object detection under the VisDrone dataset. One study [74] proposed a lightweight model, namely DMS-YOLO, with 3.77 M parameters, achieving greater overall performance as a lighter model, in comparison with other lightweight models including YOLOv8s, YOLOv10s, YOLOv11n, YOLOv12n and several transformer-based detection models.

2.6.7 Attention-Centric Spatial Modeling: YOLOv12

YOLOv12 shifts its focus on attention-centric-based frameworks through several modifications and optimizations to improve gradient flow. Firstly, they introduce a novel **Area Attention** (A^2) module to retain a large receptive field while reducing the computational complexity of attention. Secondly, they improved upon the ELAN [70] backbone architecture debating its issues of slow and unstable convergence for larger in scale models (l or x). They proposed the **Residual Efficient Layer Aggregation Network** (**R-ELAN**) as opposed to this problem. Their design leverages a residual shortcut from input to output throughout the block with a scaling factor (default to 0.01). Thirdly, they introduce **FlashAttention** to better utilize GPU memory utilization by reducing the memory access issue of attention with some removed features including positional encoding. They adjusted the MLP ratio (4 to 1.2) to balance computation between attention and feed forward networks (FFN) for greater performance. They reduced the depth of stacked blocks for optimization and finally they leveraged convolution operations more frequently to make use of their computational efficiency. YOLOv12 Figure 15 illustrates the new architectural design of R-ELAN and the improved attention mechanisms providing greater fine-grained contrast between background and other features.



(a) R-ELAN Backbone Architecture (b) Area Attention



(c) Attention Gradients for YOLOv10, v11 and v12

Figure 15: YOLOv12 Architecture Design

2.6.8 YOLO26: Edge-First Small-Target Optimized NMS-Free DFL-Free Architecture with MuSGD Training

YOLOv26 [75] is the ultimate state-of-the-art real-time object detection model for aerial imagery, with a significant number of additions for computer vision optimized for small-target detection. It was developed with the priority in optimizing for edge devices such as NVIDIA Jetson Nano and Orin. It achieves an overall comparable performance in comparison to YOLOv8, v11, v12 and v13 [76]. YOLOv26's architectural innovations includes the optional use of **End-to-End NMS-Free** inference for edge-devices, similarly to yolov10 and the removal of Distribution Focal Loss (**DFL-Free**). Previously, to achieve acceptable performance in small object detection, researchers were relying on external training techniques to improve the overall performance for small object detection. To combat this, the study adopted several advanced training strategies, including **Progressive Loss Balancing** (ProgLoss) for stable convergence and **Small-Target-Aware Label Assignment** (STAL) to significantly improve accuracy on small objects. These additions to the model, established it as superior, robust, small object detector, while maintaining edge efficiency across the board. Conversely, several other critical additions of YOLOv26 lies within its training approach. Traditional YOLO networks relied on standard AdamW or SGD variants. While they are effective, these optimizers require extensive hyperparameter fine-tuning, thus exhibiting unstable convergence in datasets with high variability such as MSCOCO [77] and aerial datasets including VisDrone [78]. The study incorporates a hybrid approach, blending standard SGD [67] with the Muon optimizer [79], a technique utilized in Large Language Model (LLM) training. The optimizer in question, **MuSGD**, utilizes Newton-Schulz iterations [80] at each gradient update to increase orthogonality, thus stabilizing the learning path for complex, high-resolution aerial features and further improving training convergence stability. These architectural and methodological improvements, rank YOLOv26 as a highly robust, state-of-the-art model for real-time detection, that can be easily integrated for aerial imagery tasks with complex scenes and small object scales. The model architecture is comprehensivelly illustrated in Figure 16. The architecture appears to be using the C3k2 module, initially introduced in YOLOv11 [73].

Moving forward, the original YOLOv26 architectural improvements were primarily adopted

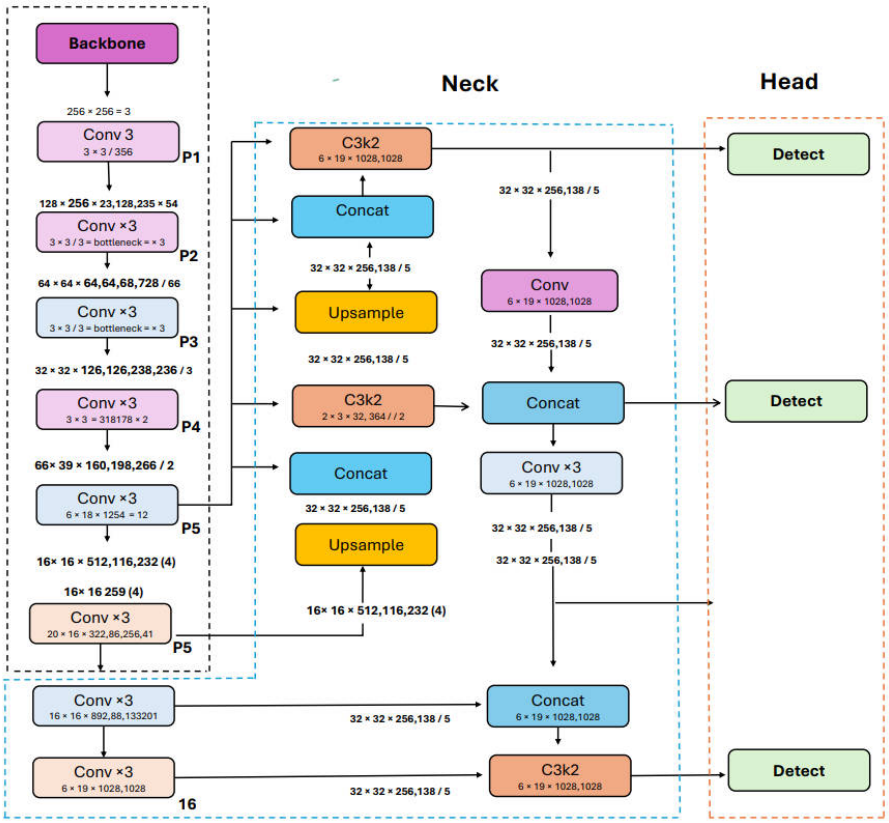


Figure 16: Comprehensive YOLOv26 Architecture Design

for detection, the ultralytics community [63] improved the capabilities of YOLOv26 by integrating a large number of computer vision task extensions. These extensions include, image classification, instance segmentation, Pose Estimation, Oriented Bounding Box Detection and introduced for the first time semantic segmentation in the YOLO series. Conversely, ultralytics broadens the context through open-vocabulary detection and instance segmentation.

2.7 Synthesis of YOLO Architectural Advancements

The progressive refinement and improvements of YOLO architectures demonstrates from dense, computationally heavy convolutional backbones, to highly optimized, attention-driven, NMS-free frameworks. As these models migrated initially from Darknet, to PyTorch, and integrated advanced training methodologies, like Dual Assignment Strategy, MuSGD, their utilization in aerial domains expanded exponentially. Each of these models were systematically encapsulated in the following Table 2, summarizing the characteristics of each major YOLO generation, in addition with MSCOCO-Test [77] evaluation results.

Table 2: Summary of Evaluated YOLO Variants: Framework, Backbone, Tasks, and mAP

Model Name	**Framework**	**Backbone**	**Computer Vision Tasks**	**mAP (50-95)% (COCO-Test)**
YOLOv4 (2020) [10]	Darknet	CSPDarknet-53	Object Detection	43.5
Scaled-YOLOv4 (2021) [58]	PyTorch	CSPDarknet-53	Object Detection	47.5
YOLOv7 (2022) [46]	PyTorch	E-ELAN	Object Detection, Instance Seg., Pose	56.8
YOLOv8 (2023) [48]	PyTorch (Ultralytics)	CSPDarknet-53 with C2f	Object Detection, Instance Seg., Pose, Classification, OBB	53.9
YOLOv9 (2024) [69]	PyTorch	G-ELAN	Object Detection, Instance Seg., Classification	55.6
YOLOv10 (2024)[71]	PyTorch	CSPNet	Object Detection	54.4
YOLOv11 (2024) [73]	PyTorch (Ultralytics)	C3k2 + SPFF & C2PSA	Object Detection, Instance Seg., Pose, Classification, OBB	54.7
YOLOv12 (2025) [49]	PyTorch	R-ELAN + A^2	Object Detection, Instance Seg., Pose, Classification, OBB	55.2
YOLO26 (2025/2026) [75]	PyTorch (Ultralytics)	CSP	Object Detection, Instance & Semantic Seg., Pose, Classification, OBB, Open-Vocabulary	57.5

These architectural improvements, reveal a relentless, mathematically rigorous optimization, strictly aligned with the demands of real-time inference and edge-deployable computer vision models. For aerial imagery applications, this evolution has systematically reduced the limitations of traditional object detectors.

Chapter 3: Computer Vision in Aerial Imagery

The consistent uprising of remote sensing and surveillance technologies has facilitated the use of Unmanned Aerial Vehicles (UAVs). Their use is particularly advantageous for applications requiring spatial analysis of vegetation ecosystems or post-disaster assessment, where satellite-based applications may lack sufficient resolution to extract granular details. The application of Computer Vision (CV) in Aerial Imagery presents a compelling case study in comparison to traditional domains. Varying altitudes and gimbal perspectives, small targets for detection and weather conditions are some of the variables taken into account when needing to develop robust CV models for detection. Moreover, UAVs are often equipped with thermal and RGB imaging sensors providing more flexible means of capturing high-resolution aerial imagery. Thermal imaging sensors enable the detection of heat signatures, which is crucial to monitor wildlife during nocturnal hours where RGB-based sensors fall short. Adopting CV methods in UAVs utilizing these imaging sensors, enhances the scope of high-fidelity data retrieval and is currently utilized in various domains, including Precision Agriculture [81], Urban Area Monitoring [82], Disaster Assessment [83], Wildlife Monitoring [84] and more. To detect objects in Aerial imagery large publicly available datasets have been developed to be utilized for training and evaluation of CV models. The goal of this thesis is to utilize one of these public datasets to train, optimize and evaluate state-of-the-art detection models and subsequently assess the accuracy of the geolocation estimation algorithm where objects are geographically localized from pixels to world maps.

3.1 Challenges of Computer Vision in Aerial Imagery

Despite these technological advances, the deployment of CV algorithms introduces a complex set of challenges. Among these, the scale problem is a profound challenge in aerial imagery where the scale of objects can vary significantly based on the altitude of the UAV, and distance to the target. For instance, at typical UAV altitudes, even prominent objects such as vehicles or pedestrians may occupy only a small fraction of an image. This renders traditional CV applications ineffective without modification. Most UAVs are equipped with wide gimbal lenses, leading to significant perspective distortion effects [3]. Additionally, aerial imagery may exhibit viewpoint variability, where fluctuating gimbal angles can alter the background context either through motion blur or dramatic shifts in the geometry of the background due to gimbal movement. An ideal gimbal angle is typically at 90 degrees focusing on the Nadir perspective, however, this in most applications is not always the case. One study developed an adaptive multi-scale vehicle and pedestrian detection framework to address this issue by automatically adjusting the convolution kernel receptive field through an Adaptive Feature Extraction Module (AFEM) [85]. The AFEM dynamically adjusts the receptive field of convolutional kernels based on object scale, enabling the model to extract features of small objects without compromising feature extraction of larger objects. While varying altitudes and gimbal perspectives introduce significant challenges, illumination inconsistencies further complicate the task of detecting objects in aerial imagery. Variations in weather conditions, cloud coverage, night and day cycles all introduce UAV Photogrammetry complexities. For instance, the presence of clouds can result in significant variations in illumination and object contrast. Moreover, haze, fog, humidity and rainy weather patterns significantly degrade image quality, leading to increased artifacts and false-positive detections during model inference. These challenges are typically mitigated through robust data augmentation techniques that simulate adverse weather patterns during

training to improve model generalizability. One study [86] introduces an effective augmentation method for geo-localization in adverse weather patterns by simulating different weather conditions and visual effects through data augmentation. Another study [87] implements novel data augmentation methodologies to simulate adverse weather conditions such as fog, sun flares, and low light conditions, thus improving the robustness of CV models in real-world scenarios.

Moving forward, due to the consistent evolution and increasing complexity of YOLO architectures, it is essential to understand the architectural and methodological choices made in this work based defined by the challenges of this domain:

- **Object Scales and Gimbal Variations**: Practical applications of aerial imagery, involves the gimbal camera to be typically lifted from the nadir point up to 45°. This results in significant object scale and perspective variations. Moreover, in varying altitudes, objects may appear significantly smaller or larger based on the drone altitude. Thus, a model optimized particularly for this task to accurately detect medium and small objects is ideal for this task. To combat this, we can train small-target-optimized models like YOLOv26 [75], however, scales may vary significantly with contrasting altitudes (e.g. 30m, 50m and 80m+). CNNs can be modified for multi-scale feature extraction and thus improve small-target detection performance without compromising medium or large-sized targets [88]. Another solution is to adopt a dual-modeling approach, by training two separate models for small and medium-sized targets.

- **Weather Conditions and Environmental Shifts**: Aerial imagery is often susceptible to weather conditions (e.g. fog, rain, snow, mist, day-night cycle) and seasonal shifts (e.g. seasonal phenology, vegetation variations, growth/erosion). These conditions can dramatically shift image luminance, visibility and contrast in color, resulting in increased noise and data variation complexity. Moreover, these variations can impact object detectors, by increasing false positive detections. To combat this, several researchers rely on augmentation methodologies without needing to modify architectures, by including complexity in datasets. This added complexity has proven to increase model performance across various scenarios. In one of my studies involving remote sensing to detect wildlife birds through UAV imagery [89], we incorporated a comprehensive augmentation method in our initial dataset and observed significant detection accuracy improvements in comparison to our initial dataset.

- **Model Input Dimensions**: Aerial imagery typically features extremely high native resolutions (2K or greater), whereas standard real-time object detectors operate on heavily downsampled inputs (such as 640×640). Resizing high-resolution drone imagery can result in catastrophic quantitative performance and severe image quality degradation. This issue becomes significantly more profound for targets occupying less than 1% of the overall image resolution. There are several methods to mitigate this, either by integrating Slicing Aided Hyper Inference SAHI [90], however, this introduces a new problem that is well documented in one of my studies for small target detection of wildlife birds [91]. Another solution is to utilize a larger receptive field for high-resolution imagery, however, this exacerbates the Receptive Field Attention (RFA) problem.

- **Receptive Field Attention (RFA) Problem**: Standard DNNs leveraging advanced attention mechanisms [49, 92] are structurally constrained to medium-sized receptive fields

or localized context windows. When applied within larger receptive fields, uniform spatial attention mechanisms routinely fail due to the mathematical dilution of fine-grained features. The attention maps undergo an adverse "oversmoothing" effect, flattening the activation response and submerging objects that account for less than 1% of the scene. In short, models leveraging attention-mechanisms when trained in higher resolution imagery, the RFA problem may significantly impact and reduce performance in complex datasets with high variability. Several studies actively combat the RFA problem through guided, dynamic, or adaptive attention techniques tailored for aerial and UAV imagery [93, 94].

A large portion of these issues can be resolved by introducing complex multi-domain datasets with high variability in weather conditions and seasonal shifts. Several public datasets have been developed for researchers to utilize as standard benchmarks in aerial imagery. The next section expands on several of these datasets that resolve the issues of weather conditions, object scales and UAV perspective shifts.

3.2 Public Domain CV Datasets for Aerial Imagery

To detect objects in Aerial imagery large publicly available datasets have been developed to train and evaluate CV models. These datasets are often collected using high-end commercial drones, capturing high-fidelity data across various environments, altitudes and gimbal angles. Each dataset is designed to address specific computer vision tasks (detection, semantic segmentation, object tracking) and challenges associated with aerial imagery, such as small object detection, variations in illumination conditions, and perspective distortions. For this thesis, several datasets were gathered to train and evaluate state-of-the-art detection and tracking models. Table 3 provides a comprehensive overview of these datasets, detailing their specific tasks, demographics and challenges.

Table 3: UAV-Based Computer Vision Datasets

Dataset Name	Task	Demographics	Key Notes
VisDrone [78]	Object detection, single-object tracking, and multi-object tracking.	288 video clips, 10,209 static images, 10 object categories (e.g., pedestrian, car, bicycle).	Captured over 14 Chinese cities. Challenges include high object density and minute target sizes caused by varying flight altitudes.
UAVid [95]	Semantic segmentation.	300 densely annotated high-resolution images from 42 video sequences, 8 semantic classes.	High-resolution, oblique-view video sequences. Key challenges are large scale variation and complex urban scene understanding.
FloodNet [96]	Post-disaster scene understanding, semantic segmentation, and Visual Question Answering (VQA).	2,343 high-resolution images with 10 pixel-level semantic categories (e.g., flooded vs. non-flooded structures).	Captured after Hurricane Harvey for damage assessment. Challenges include domain-specific generalization for disaster response.
UAVDT [97]	Vehicle detection and tracking.	100 video sequences (≈80,000 frames) focusing on 3 vehicle categories (car, truck, bus).	Incorporates complex scenarios, including weather variations, varying flying altitudes, and multiple camera angles.

Moving forward, to effectively detect vehicles and pedestrians across varying environmental conditions and perspectives, this thesis initially utilized the VisDrone dataset [78] to train and evaluate state-of-the-art YOLO (You Only Look Once) [7] models, built for real-time object detection. The VisDrone dataset stands as the largest and most comprehensive benchmark for aerial object detection, containing over 10,209 annotated static images and 288 video clips captured from drones flying over 14 Chinese cities. It features 10 distinct object categories including pedestrians, vehicles, bicycles, and various other targets often encountered in urban and rural environments. The dataset is specifically designed to address the unique challenges posed by aerial imagery, including extreme scale variations, high object densities, and small target sizes resulting from varying flight altitudes. Training was conducted on the VisDrone dataset with careful attention to the unique characteristics of aerial imagery. Given that VisDrone contains high-resolution aerial imagery in addition with YOLO models automatically scaling input resolutions to match the training configurations. Resizing high-resolution imagery (typically 2k or higher) to much lower dimensions (640×640) will result in major image quality degradation which is particularly more profound to smaller objects occupying very few pixels relative to the whole image. To mitigate these resolution constraints without impacting small object detection performance, careful consideration in training configurations, including the optimization of underlying network structures.

3.3 Comprehensive Analysis of the VisDrone Dataset

The VisDrone2019 dataset serves as a foundational benchmark for object detection and tracking in aerial imagery. Captured across diverse urban and rural environments under varying lighting and weather conditions, the dataset presents a highly complex multi-domain set of images ideal for aerial imagery applications. Unlike ground-level datasets, aerial imagery introduces unique challenges, including extreme perspective distortions, object scale variability, and drastic variations in object density counts, depending on uav altitude and gimbal pitch.

A granular statistical analysis of the dataset reveals several critical characteristics that heavily influence model performance. First, the class distribution exhibits a severe long-tail imbalance, Figure 17. Categories such as pedestrians and standard vehicles dominate bounding box instances, while other classes like tricycle, awning-tricycle and others are significantly underrepresented. This imbalance persists consistently across the training, validation, and testing splits, requiring robust augmentation strategies to prevent model bias towards majority classes.

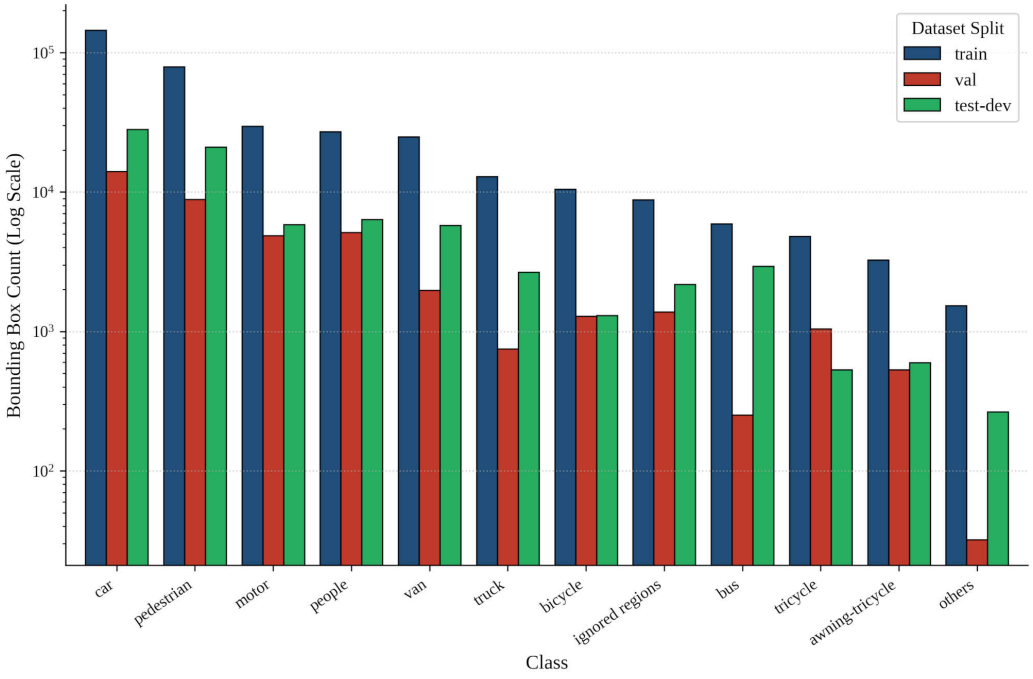


Figure 17: Granular breakdown of class occurrences stratified by dataset splits.

Second, the dataset demonstrates profound object scale variation. The UAV was operated at variable altitudes and gimbal angles where several objects captured within the dataset occupy only a fraction of the scene, while others objects were larger and distinguishable. An analytical breakdown of bounding box dimensions shows an overwhelming majority of instances falling into the "small" object category, frequently occupying bounding boxes of $\leq 32 \times 32$ pixels or smaller, Figure 18. Labels with really small boundaries severely limit any spatial information that can be extracted, making localization and classification difficult. Moreover, a high concentration of these tiny features

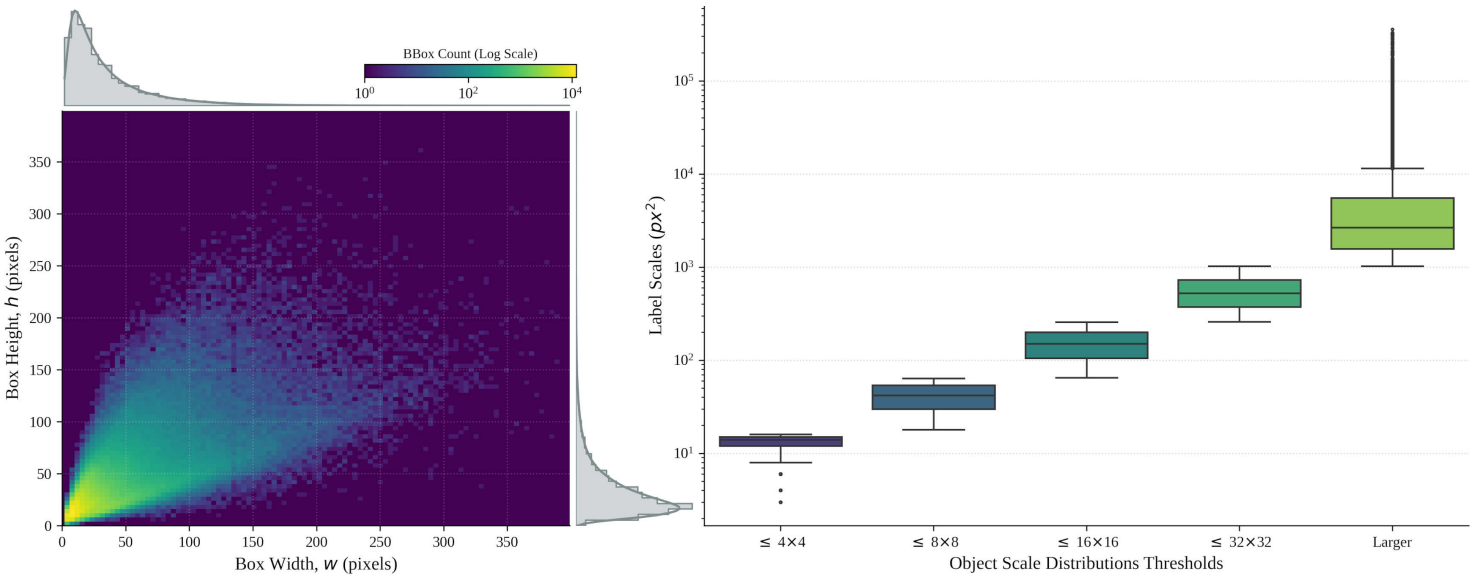


Figure 18: Granular Analysis of Bounding Box Dimensions

3.4 Computer Vision Performance Metrics

Computer Vision models are typically evaluated using a structural framework derived from the traditional Confusion Matrix, by comparing ground-truth bounding boxes into True Positive (TP), False Positive (FP), and False Negative (FN). Conversely, for object detection, these classification metrics are typically calculated based on an Intersection over Union (IoU) threshold, defining the overlap percentage between the prediction and ground-truth bounding boxes. From these parameters, we can calculate fundamental analysis metrics including **Precision** ($P = \frac{\text{TP}}{\text{TP+FP}}$) and **Recall** ($R = \frac{\text{TP}}{\text{TP+FN}}$). Precision indicates the proportion of true targets among all predictions, while Recall measures the detector's capability to find all ground-truth objects within each frame. The primary benchmark utilized to evaluate and rank detector performance is the **mean Average Precision** (mAP), that is calculated using Precision-Recall across all class categories. For this analysis, we report this metric using two criteria:

- mAP_{50}: Evaluates precision and recall at a fixed IoU threshold of 0.50, capturing the coarse performance of the model.
- mAP_{50-95}: Evaluates precision and recall by averaging performance of progressive IoU thresholds from 0.50 to 0.95, with a step size of 0.05. This enforces a condensed and more accurate measure to evaluate overall detector performance.

Finally, to provide a balanced geometric mean that accounts both for False Positives and missed detections simultaneously, we can utilize the **F1-Score**:

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{1}$$

Conversely, because the VisDrone dataset consists of altitude variations and high object densities, evaluating purely through global mAP scores is not sufficient and can hide specific architectural benefits and limitations. To reveal these, performance metrics can be broken down by object scales:

- mAP_{Small}: Targets spanning an area under 32×32 pixels.
- mAP_{Medium}: Targets spanning an area between 32×32 and 96×96 pixels.
- mAP_{Large}: Targets spanning an area over 96×96 pixels.

In semantic segmentation tasks, due to its nature, all metrics are calculated at a pixel level rather than using bounding box coordinates. For instance, IoU, similarly to object detection, it measures the predicted mask's overlap between the ground-truth mask. Mathematically, the formula is defined using the concepts of intersection ($\cap$) and set Union ($\cup$):

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

- A: The set of pixels predicted by the model, either class-specific or coarse.
- B: The set of ground-truth pixels from the dataset, either class-specific or coarse.
- $|\cdot|$: The total count of pixels contained within the combined spatial sets.

3.5 YOLO Model Performance on VisDrone dataset

To identify the best performing real-time object detector for Aerial Imagery-driven tasks, several state-of-the-art YOLO variants were evaluated under the VisDrone validation set. Each model was trained using the same hyper-parameters as listed in table 4.

Table 4: Training Hyper-parameters for a standard training process for each YOLO Variant

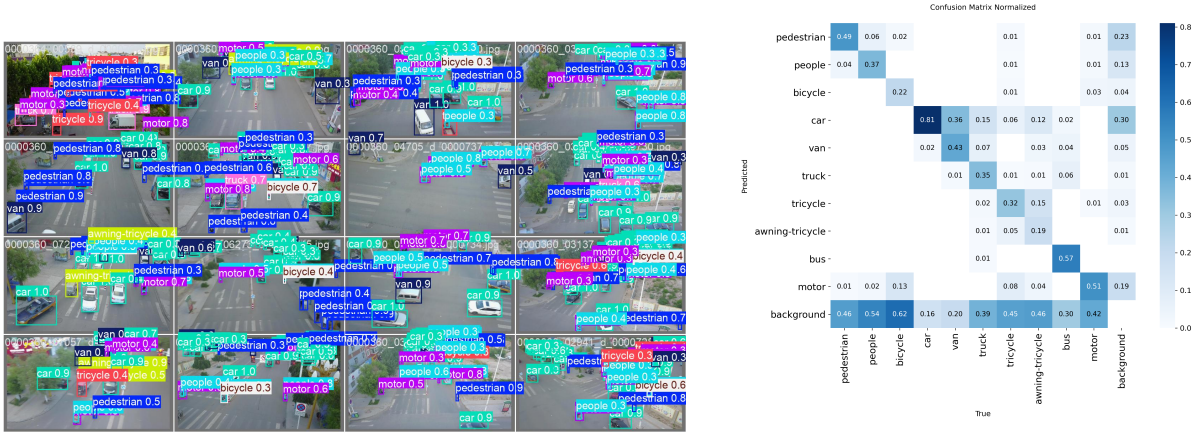
Hyper-parameter	**Value**	**Description**
data	dataset	Path to the configuration file (YAML)
epochs	50	Number of training iterations
imgsz	1280	Input image resolution (pixels)
batch	6	Training batch size
GPU	NVIDIA RTX 4090 (24GB)	Computational hardware utilized

Table 5 summarizes the performance of different YOLO variants that were trained in this study for $epochs = 50$ and $imgsz = 1280$ using the VisDrone dataset with no augmentation.

Model Name	**mAP (50)**	**mAP (50-95)**	**F1-Score**	**mAP Small**	**mAP Medium**	**mAP Large**	**Eval Time (ms)**
yolov8m	0.5837	0.3669	0.6099	0.2424	0.2026	0.6536	7.42
yolov9m	0.5929	0.3728	0.6189	0.2509	0.2181	0.6547	8.30
yolo11m	0.5929	0.3737	0.6148	0.2484	0.2097	0.6591	7.18
yolo12m	0.5867	0.3705	0.6108	0.2422	0.2124	0.6581	11.54
yolo26m	0.5883	0.3729	0.6089	0.2593	0.2186	0.6581	6.86

Table 5: Performance Summary of Medium-Scale YOLO Variants on VisDrone

An analysis of these results demonstrates a direct correlation between the architectural variations of each detector under the complexities of the VisDrone dataset. The baseline performance of yolov8m reflects the structural limitations of the C2f block when it's subjected to extreme scale variations, it performed the worst ($mAP_{50-95} = 0.3669$) compared to the newer architectures. YOLOv9 and YOLOv11 as the direct successors of yolov8 had among the highest mAP (0.3728 and 0.3737 respectively) and F1 Scores (0.6189 and 0.6148 respectively). YOLOv12, due to the Area Attention A^2 module, the lighter R-ELAN backbone and the primary complexity of the RFA problem that was reported previously. Conversely, YOLOv26m achieved the best balance between structural efficiency achieving the best evaluation time in milliseconds ($evaltime = 6.86ms$) and small-target sensitivity, performing adequately well in small and medium-sized targets (mAP Small = 0.2593, mAP Medium = 0.2186 respectively). The illustration of Figure 19a, demonstrates a 4×4 batch of images from the VisDrone validation set with YOLO26m detections along with confidence values.



(a) Batch of images from VisDrone Validation set and YOLO26-e2e model Detections.

(b) Confusion Matrix of YOLO26m-e2e

Figure 19: YOLO26m-e2e Results

3.6 Architectural Improvements for Small Target Detection

Through the systematic review of various YOLO evolutions and the in-depth analysis on the challenges of Computer Vision in Aerial Imagery, several architectural improvements can be integrated to further advance precision and accuracy for challenging datasets such as VisDrone 2019 [78].

3.6.1 Global Context through Area Attention and Partial Self Attention

Initially, the YOLOv11 study introduced the C2PSA convolution block that integrates a 2×2 convolution and the Partial Self-Attention module at the final stage of the backbone, positioned immediately following the Spatial Pyramid Pooling (SPP) module [73]. Subsequently, the YOLOv12 study proposed the Area Attention (A^2) module, an innovative mechanism that refines attention gradients to extract finer spatial details as per the paper [49]. While these components demonstrated significant potential, they were previously integrated across several parts of the network using an unoptimized C2f module that limited its efficiency. The previous evaluations under the VisDrone dataset in Table 5 reveal a performance gap between

YOLOv9/v11 and YOLOv12. This likely stems from several issues including the Receptive Field Attention (RFA) problem, inefficient optimizations for small target detection, and the architectural limitations of the C2f blocks utilized in predecessor models (v7, v8, v9, v10).

To address these issues, this thesis proposed the **A2C2PSA-SE** module that merges the YOLOv12 Area Attention Block [49] with the YOLOv11 C2PSA module [73]. This module aims to refine feature maps using attention mechanisms and is placed at the final stage of the backbone for refined global context. This addition alone increased the overall mAP0.5:0.95 by 7.25% in relation to YOLOv26n. While these components themselves demonstrate significant potential, their integration may aggravate the RFA problem, where models will fail to adjust to finer edges, thus, impacting performance on small object detection. Moving forward, this was combated by adopting an Squeeze-and-Excitation (SE) Block [98] mechanism within **A2C2SPA** to

3.6.2 Multi-Scale Adaptation

Several studies [93, 99] have incorporated similar multi-scale feature extraction techniques to optimize for small object detection. To adopt this multi-scale feature extraction, this thesis focused primarily in the implementation of YOLO-MST [88] where they replaced the SPPF module with a Multi-scale Feature Aggregation (MSFA) module. This addition leverages dilated convolutions with rates of 1, 3, and 5 to extract features across different scales that are conversely summarized into a single feature map using a 1×1 convolution:

$$F_{MSFA} = Conv(Conv_1(F) + Conv_2(F) + Conv_3(F)) \tag{3}$$

Where Conv_x represents convolution with rate 1, 3 and 5 respectively and F_{MSFA} the convoluted feature map of the three convolutions.

For their study, the integration of the MSFA module yielded significant performance gains of 3.7% in mAP0.5 and 2.4% in mAP0.5:0.95. Furthermore, the study incorporated a Dynamic Head transformer (DyHead) [100] as the primary detector providing additional improvements of 2.5% and 2.0% in mAP0.5 and mAP0.5:0.95, respectively. These modifications significantly boosted the overall performance relative to other YOLO models under the SIRST dataset [101], from a baseline YOLOv5s achieving 0.693 mAP50 and 0.297 mAP50-95 to a staggering 0.964 mAP50 and 0.475 mAP50-95.

Another alternative methodology to optimize models for small target detection, several architectural improvements can be made across YOLO architectures starting from YOLOv4-Scaled and above. To prioritize in small-scale predictions, the Feature Pyramid Network (FPN) of each architecture can be modified by replacing any P5 head with a P2 head, while maintaining P3 and P4. The P5 head prioritizes large occlusion bounding boxes, whereas, lower numerically heads, prioritize smaller in scale bounding boxes. Each FPN scale extracts features of different scales, a P2 head divides the image into small 4×4 pixels (16), while a P5 head, creates 32×32 pixels (1024), leveraging a P2 head increases precision for small target detection but significantly raises computational overhead, making inference slower and unreliable for real-time applications.

Chapter 4: Geolocation Estimation Pipeline

Building upon the detection of objects, the primary goal of this thesis is to develop a fully state-of-the-art UAV-based Geo-referencing framework where local pixels from an image plane are projected onto Global Coordinate Systems (e.g. WGS-84 [102]). **Geolocation Estimation** is a newly defined field in computer vision that is comprehensively explored within this thesis. The integration of computer vision enables the full automation of this process by utilizing model predictions and UAV telemetry data from onboard sensors. Modern UAVs are equipped with state-of-the-art sensors and equipment providing highly precise data. Moving forward, it is crucial to understand and define the technologies of these sensors. To project local pixels to GPS coordinates, we utilize intrinsic and extrinsic camera parameters. The next section expands more thoroughly on the telemetry data used for Geolocation Estimation.

4.1 Study UAV and Intrinsic/Extrinsic Parameters

To systematically map a localized pixel coordinate (u, v) from a two-dimensional image plane to a three-dimensional coordinate system, the framework must utilize both the internal and external geometric properties of the camera lens. Intrinsic parameters refers to the internal optical geometric properties of a camera lens, such as focal length, principal point and skew, while extrinsic parameters, refers to camera position and orientation in 3D space relative to the real world. Intrinsic parameters are specific to camera hardware and typically remain constant unless the lens settings change (e.g. zoom, crop). Extrinsic parameters shift constantly for UAV-base applications due to movement, thus, precise sensors need to be utilized for accurate GPS estimation of target objects in an image. Moving forward, for the practical validation of this thesis, a commercial UAV was utilized (DJI Matrice 30 Thermal), equipped with an integrated multi-sensor payload containing a high-resolution zoom, wide-angle and thermal lenses, finally it includes several other sensors detailing the gimbal orientation and position relative to real world coordinates. For this study, we primarily utilized the wide-angle lens and its subsequent intrinsic parameters as thoroughly described in Table 6.

Table 6: DJI Matrice 30T Onboard Multi-Sensor Intrinsic Specifications

Specification Parameter	**Wide-Angle Camera**	**Zoom Camera**	**Thermal Camera**
Sensor Type	1/2-inch CMOS	1/2-inch CMOS	Uncooled VOx Microbolometer
Active Pixels	12 MP	48 MP	0.31 MP
Maximum Resolution	4000×3000 pixels	8000×6000 pixels	640×512 pixels (1280×1024 Super-Res)
Diagonal FOV	84°	60.7° to 2.2°	61°
Equivalent Focal Length	24 mm	113–403 mm	40 mm
Physical Focal Length (f)	4.5 mm	21–75 mm	Approx. 9.1 mm
Aperture	$f/2.8$	$f/2.8$ to $f/4.8$	$f/1.0$
Intrinsic State	Static Matrix ($\mathbf{K}_{\mathbf{wide}}$)	Matrix ($\mathbf{K}_{\text{dynamic}}$)	Matrix ($\mathbf{K}_{\text{thermal}}$)

As structured in Table 6, the hardware within each payload dictates how the intrinsic calibration matrices should be constructed during model deployment. For the wide-angle and thermal sensors, the fixed focal lengths maintain relatively uniform pixel scaling factors (f_x, f_y) across all captured frames, facilitating stable geometric projection. However, the thermal lens introduces specific geometric aberrations—such as a pincushion effect—resulting in significant

radial distortion and image skew. Conversely, the zoom camera presents a complex architectural challenge for real-time geolocation estimation due to its dynamic optical focal range spanning from 113 mm to 403 mm. Due to these discrepancies, the wide-angle lens was primarily leveraged for the baseline geolocation estimation. The precise intrinsic parameters of the wide-angle lens were extracted via DJI Terra, a proprietary photogrammetry application, including the Radial and Tangential Distortion Coefficients. Each of these parameters are detailed in Table 7:

Table 7: Camera Intrinsic and Distortion Parameters

Parameter	**Value**	**Description**
	Sensor Dimensions	
`SENSOR_W`	4000	Native horizontal resolution of the sensor array.
`SENSOR_H`	3000	Native vertical resolution of the sensor array.
	Sensor Angular Dimensions	
`FOV_D_DEG`	84.00°	Diagonal Field of View (FOV_d).
`FOV_H_DEG`	71.53°	Horizontal Field of View (FOV_h).
`FOV_V_DEG`	56.30°	Vertical Field of View (FOV_v).
	Sensor Principal Point	
`PP_43_CX` (c_x)	2000.0	X-coordinate of the true optical principal point (dead center of the 4:3 frame).
`PP_43_CY` (c_y)	1500.0	Y-coordinate of the true optical principal point (dead center of the 4:3 frame).
	Sensor Focal Length expressed in pixels	
`FOCAL_FX` (f_x)	2742.86	Focal length expressed in pixel units along the horizontal axis.
`FOCAL_FY` (f_y)	2742.86	Focal length expressed in pixel units along the vertical axis.
	Radial Distortion Coefficients (4:3 Lens)	
`K1` (k_1)	0.175877	First-order radial distortion coefficient (Brown-Conrady model).
`K2` (k_2)	-0.425057	Second-order radial distortion coefficient.
`K3` (k_3)	0.213170	Third-order radial distortion coefficient.
	Tangential Distortion Coefficients (4:3 Lens)	
`P1` (p_1)	-0.000430	First-order tangential distortion coefficient representing decentering (lens-sensor plane misalignment).
`P2` (p_2)	0.000268	Second-order tangential distortion coefficient.

It is important to note that DJI does not address a specific issue where the video files are asymmetrically cropped to a 16:9 aspect ratio to match specific resolutions (3840×2160). This asymmetric crop shifts the true angular spans and principal points of the image. Moreover, when employing Geolocation Estimation, the estimated GPS coordinates will be heavily offset due to these variable shifts. To ensure precise Geolocation, these asymmetries must be established mathematically. The next section thoroughly expands on these discrepancies to effectively

resolve and accurately perform geolocation estimation

4.2 Asymmetric Sensor Cropping and Principal Point Displacement

The native 4:3 wide-angle sensor corresponds to the geometric center of the sensor array. For a native resolution of 4000×3000 pixels, the coordinates of the principal point, $C_x^{4:3}$ and $C_y^{4:3}$, are defined as:

$$C_x^{4:3} = \frac{4000}{2} = 2000 \text{ px}, \qquad C_y^{4:3} = \frac{3000}{2} = 1500 \text{ px}$$

However, during video acquisition, an asymmetric crop is initially performed to achieve a 16:9 aspect ratio (3840×2160 pixels). By aligning the 4:3 image with the cropped image, reveals the precise pixel bounds of this asymmetric crop window, thus, the true optical principal point is displaced relative to the local coordinate system of the 16:9 crop. The effective principal point within the 16:9 frame is expressed as:

$$C_x^{16:9} = C_x^{4:3} - x_{\text{left}} = 2000 - 100 = 1900 \text{ px}$$

$$C_y^{16:9} = C_y^{4:3} - y_{\text{top}} = 1500 - 434 = 1066 \text{ px}$$

While the native 4:3 focal length is 2742.857 px, the 16:9 frame undergoes a scaling shift during pipeline processing, resulting in an effective focal length of $f_{16:9} = 2633.143$ px. Because the crop window is off-center relative to the optical axis and the crop itself is asymmetric, the angular boundaries of the 16:9 frame are displaced. To maintain geometric fidelity when calculating ground projections from the 16:9 image, the angular boundaries must be computed utilizing this specific 16:9 focal length.

Applying a Cartesian sign convention (where rightward and upward shifts denote positive values relative to the optical axis), the angular boundaries of the 16:9 field of view are calculated as $\tan\theta = \frac{\Delta \text{px}}{f_{\text{DJI}}}$. The angular tangent boundaries are defined by the set of values for the left, right, top, and bottom edges:

$$\tan\theta_{x,\text{left}} = \frac{100 - 2000}{2742.857} = \frac{-1900}{2742.857} \approx -0.6927 \implies \theta \approx -34.71^\circ$$

$$\tan\theta_{x,\text{right}} = \frac{3940 - 2000}{2742.857} = \frac{+1940}{2742.857} \approx +0.7073 \implies \theta \approx +35.27^\circ$$

$$\tan\theta_{y,\text{top}} = \frac{1500 - 434}{2742.857} = \frac{+1066}{2742.857} \approx +0.3886 \implies \theta \approx +21.24^\circ$$

$$\tan\theta_{y,\text{bottom}} = \frac{1500 - 2594}{2742.857} = \frac{-1094}{2742.857} \approx -0.3988 \implies \theta \approx -21.74^\circ$$

This asymmetric configuration yields a total horizontal FOV of approximately 69.98° and a vertical FOV of 42.98°. Previous pipeline iterations erroneously utilized a scaled 16:9 focal length ($f \approx 2633.14$ px), which artificially inflated the angular boundaries (yielding 72.19° HFOV and 44.60° VFOV) and resulted in a significantly over-extrapolated ground footprint, therefore, it was considered within the framework to optimize and improve geolocation accuracy.

4.3 Pipeline Implementation and Lens Distortion Modeling

Within the analytical pipeline, these asymmetric boundaries are applied exclusively to 16:9 geometries. For a given pixel (u, v) in the 16:9 frame (where $v = 0$ defines the top edge), the pixel-to-angle mapping is established via linear interpolation of the boundary tangents:

$$\tan\theta_x(u) = \tan\theta_{x,\text{left}} + \frac{u}{w}\left(\tan\theta_{x,\text{right}} - \tan\theta_{x,\text{left}}\right) \tag{4}$$

$$\tan\theta_y(v) = \tan\theta_{y,\text{top}} + \frac{v}{h}\left(\tan\theta_{y,\text{bottom}} - \tan\theta_{y,\text{top}}\right) \tag{5}$$

To ensure high-fidelity spatial mapping, raw pixel coordinates must be corrected for optical aberrations prior to projection. A 5-parameter Brown-Conrady distortion model (accounting for both radial and tangential distortion) is applied. The distortion coefficients, explicitly extracted via DJI Terra calibration, are presented in Table 8.

Table 8: Brown-Conrady Lens Distortion Coefficients (DJI Terra)

Distortion Type	**Parameter**	**Value**
Radial	K_1	0.1758772
Radial	K_2	−0.42505682
Radial	K_3	0.21317029
Tangential	P_1	−0.00043036
Tangential	P_2	0.00026796

Given the normalized image coordinates x and y, and the squared radial distance $r^2 = x^2 + y^2$, the distorted coordinates (x_d, y_d) are computed as:

$$x_d = x\left(1 + K_1r^2 + K_2r^4 + K_3r^6\right) + \left[2P_1xy + P_2\left(r^2 + 2x^2\right)\right] \tag{6}$$

$$y_d = y\left(1 + K_1r^2 + K_2r^4 + K_3r^6\right) + \left[P_1\left(r^2 + 2y^2\right) + 2P_2xy\right] \tag{7}$$

Applying these empirically derived coefficients effectively models the inherent barrel/pincushion distortion of the wide-angle lens. This correction slightly compresses the corner projections, precisely aligning the digital model with the physical ground footprint and resolving previously observed boundary over-extensions.

4.3.1 Extrinsic Parameter and Hardware Constraints

This spatial transformation relies on three core hardware-derived parameters to translate the image plane coordinates into real-world geographic positions:

1. **GNSS-RTK Positioning:** Real-Time Kinematic GPS provides centimeter-level absolute positioning of the UAV in global coordinates, minimizing the baseline tracking error of the camera center.

2. **Gimbal Orientation Angles:** The pitch, roll, and yaw angles of the stabilized camera gimbal define the rotation matrix required to align the image plane's line of sight with the global coordinate system.

3. **Camera Altitude:** The relative altitude of the UAV above ground level.

The absolute position of the aerial platform is conventionally determined via Global Navigation Satellite Systems (GNSS), which calculate position through satellite trilateration. However, standard GNSS accuracy is inherently limited to a range of ±1.5 to 5 meters due to atmospheric delays, satellite clock drift [103], and Ephemeris errors [104]. Real-Time Kinematic (GNSS-RTK) technology mitigates these limitations by introducing a stationary base station at a precisely surveyed location alongside the mobile UAV rover. By transmitting live differential carrier-phase corrections via Ultra High Frequency (UHF) link or Network Transport of RTCM via Internet Protocol (NTRIP), common-mode atmospheric and timing errors are eliminated, resulting centimeter-level positioning accuracy. Figure 20 physically illustrates the GNSS-RTK process when employed with moving vehicles

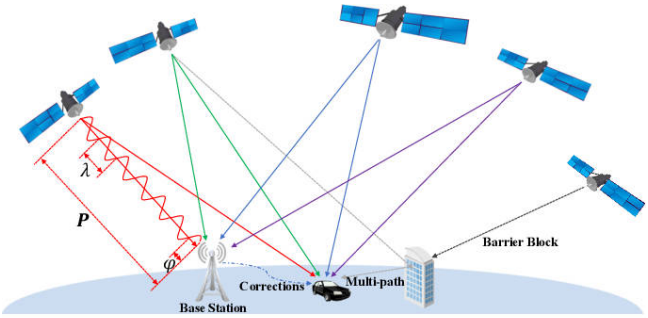


Figure 20: GNSS-RTK Process Leveraging 4-5 satellite systems combined with a grounded base station to calculate highly precise GPS coordinates of given device

Despite its advantages, GNSS-RTK deployment is frequently constrained by infrastructure. In remote, unmapped, or contested geographic locations, network RTK is often unavailable due to a deficiency in nearby Continuously Operating Reference Stations (CORS). For enterprise platforms such as the DJI Matrice 30T, exploiting RTK precision necessitates the deployment of a local proprietary base station, specifically the DJI D-RTK 2 High Precision GNSS Mobile Station. Because our experimental setup lacked this specialized local hardware infrastructure to broadcast differential corrections, the payload automatically reverted to standard GNSS positioning. Consequently, the absolute positional accuracy of the projection pipeline degrades to standard meter-level margins, which represents a critical hardware constraint for the final geolocation estimation accuracy.

4.4 Projection of 2D Image Plane to Global Coordinate System

The geolocation pipeline operates as follows: given a detected object at pixel coordinate (u, v) in an image, we compute the corresponding ray in 3D space using the camera's intrinsic parameters and the drone orientations (Gimbal Pitch/Yaw/Roll) (Figure 21).

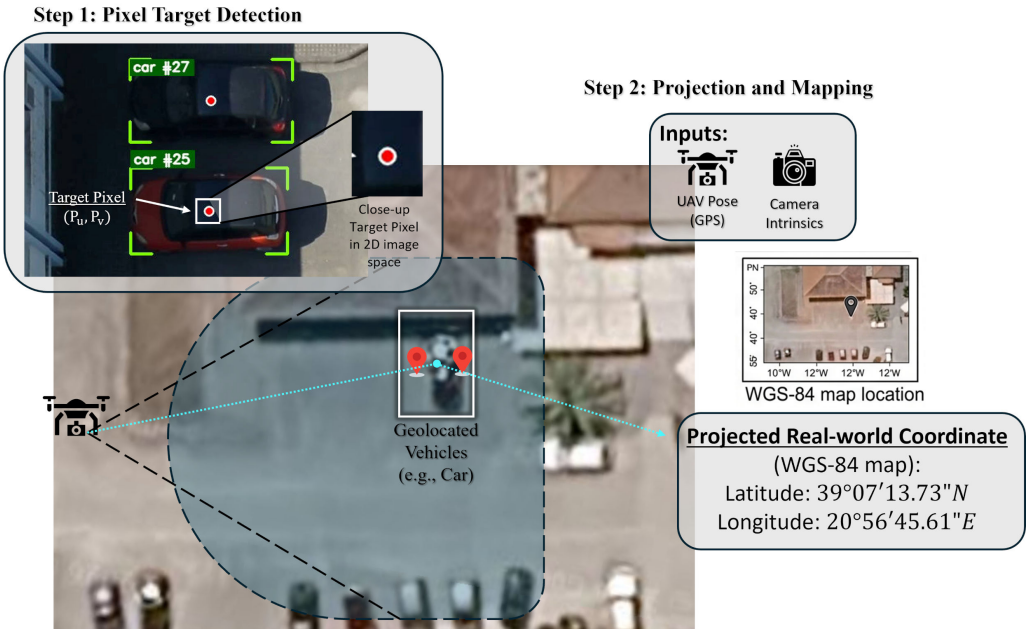


Figure 21: Geolocation Pipeline: Projection of Target Pixels in UAV Images

When projecting pixels into a The fundamental challenge in this approach is that depth (distance along the ray) must be inferred from context, as a single 2D image provides no direct depth measurement. To project a ray from local camera coordinate (pixels) into a geographical coordinate systems (Latitude, Longitude), a initial geodetic spatial reference point must be established. This reference point is formulated as the UAV's current coordinate parameter vector, provided by onboard telemetry data:

$$\boldsymbol{GPS}_{uav} = \begin{bmatrix} \phi_{uav} \\ \lambda_{uav} \end{bmatrix}$$

Where ϕ_{uav} represents the initial geodetic latitude and λ_{uav} the geodetic longitude of the UAV. This spatial reference parameter also defined as Global Positioning System (GPS) Coordinate, acts as an anchor coordinate vector for the localized scene.

To effectively project the pixels of the UAV's lens into the Global Coordinate System, we need to follow a four-step calculation:

- **Target Pixel Displacement:** Establishes the localized target coordinates within the image plane by calculating the pixel offsets $(\Delta u, \Delta v)$ relative to the Principal Coordinate Point (PCP). By inverting the camera intrinsic matrix ($\mathbf{K}$), these raw pixel offsets are mapped into a normalized camera ray vector ($\mathbf{v}_{\text{ray}}$), which naturally corrects for the structural optical axis pointing shifts introduced by the 16:9 widescreen crop.

- **3-Dimensional Rotational Matrices:** Transforms the localized camera ray vector ($\mathbf{v}_{\text{ray}}$) from the moving camera coordinate system into a global reference orientation. By leveraging the high-frequency telemetry angles of pitch (ψ) and yaw (ϕ) reported by the active 3-axis stabilizing gimbal (with roll isolated as zero), the framework constructs a composite rotation matrix ($\mathbf{R}_{\text{gimbal}}$) to compute the world ray vector ($\mathbf{V}_{\text{world}}$).

- **Ray-to-Ground Intersection:** Resolves the missing structural depth component of the single 2D image plane by projecting the world ray vector until it intersects the terrain.

This calculates a physical line-of-sight distance scalar (s) using either the drone's instantaneous altitude (H) over a flat horizontal ground plane baseline ($z = 0$), or through an iterative ray-tracing loop that samples heights along the vector until it pierces a 2D Digital Elevation Model (DEM) surface.

- **Scalar Metric Scaling to Geodetic Degrees:** Converts the true horizontal metric ground displacements ($\Delta X, \Delta Y$) in meters derived from the scaled ray into angular geodetic shifts. Using the localized meridional (R_M) and prime vertical (R_N) radii of curvature calculated from the WGS-84 reference ellipsoid at the current latitude, these linear offsets are mapped to fractional degrees and appended to the anchor coordinates ($\mathbf{GPS}_{\text{uav}}$) to output the target's absolute geographical coordinates ($\mathbf{GPS}_{\text{tgt}}$).

4.4.1 Established Image Principal Point Coordinates

To map a target pixel from the image plane $P(u, v)$ to GPS_{tgt}, the Principal Coordinate Point (PCP) also known as the center point of the image plane, is established as the origin point (0, 0). Within the framework itself, PCP is determined through the image dimensions as described previously in the updated intrinsic parameters, assuming the intrinsic parameters of DJI M30T apply for given use-case:

$$PCP = \begin{pmatrix} C_x & C_y \end{pmatrix} = \begin{cases} (2000,\ 1500), & \text{if image is 4:3,} \\ (2009,\ 1517), & \text{if image is 16:9.} \end{cases}$$

4.4.2 Target Pixel Displacement from Principal Point

For any given pixel P(u, v), we first determine the point's displacement P(Δ_x, Δ_y) from PCP. A visual illustration of the target pixel displacement relative to the Principal Point of the UAV image is demonstrated in the below Figure 22.

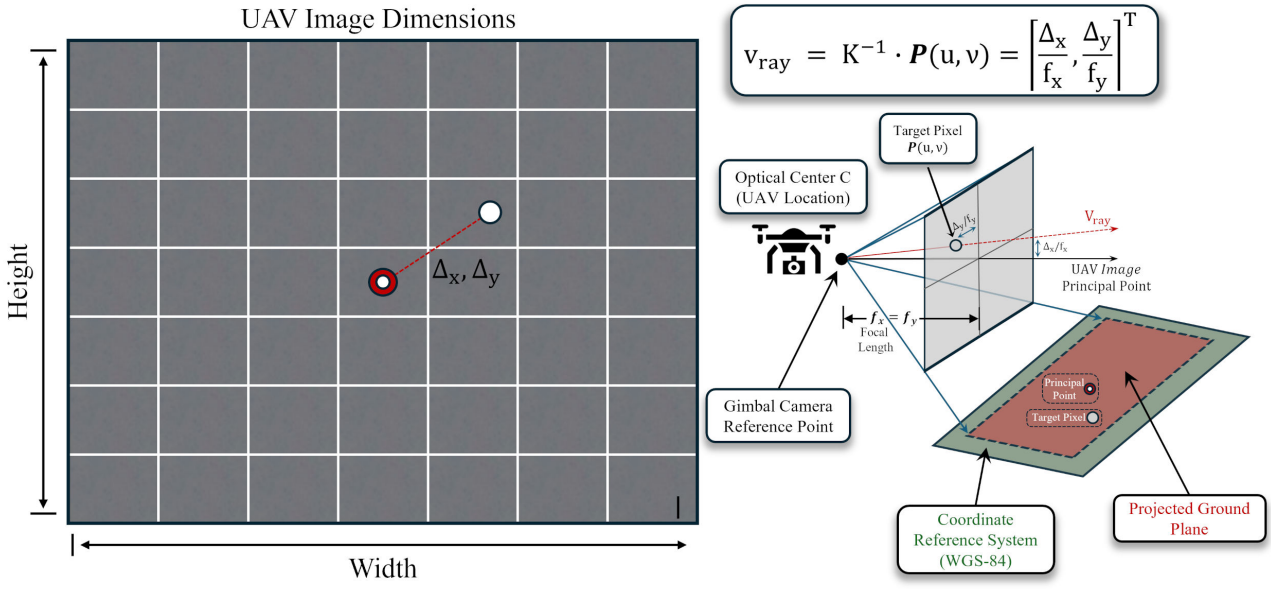


Figure 22: Target Pixel Displacement from the Principal Point

To compute the displacement, a Camera Intrinsic Matrix (K) is built and populated with the drone camera intrinsic parameters:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

Where f_x and f_y the focal point lengths of x and y axis respectively, the value '1' serves as an identity and scales for the z axis and C_x, C_y the principal point. From the previous calculations of the intrinsic properties of the camera lens, we can verify that f_x and f_y are identical and equal to $f_x = f_y = 2776.54$ pixels, whereas for PCP it is determined based on the type of image (typically: $\frac{Width}{2}, \frac{Height}{2}$). Using K, we translate the raw pixel offsets into physical GPS coordinates relative to GPS_{uav}. Given a detected object's pixel center at p = $\begin{bmatrix} u & v & 1 \end{bmatrix}^T$ in homogeneous coordinates, we compute its physical displacement vector relative to the asymmetric principal center (C_x, C_y). By inverting the intrinsic calibration matrix $\mathbf{K}$, we map this pixel location into a normalized Camera Ray Vector $\mathbf{V}_{ray}$.

$$\mathbf{V_{ray}} = \mathbf{K}^{-1} \cdot \mathbf{P}(\mathbf{u}, \mathbf{v}) = \begin{bmatrix} f_x & 0 & C_x \\ 0 & f_y & C_y \\ 0 & 0 & 1 \end{bmatrix}^{-1} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{u - C_x}{f_x} \\ \frac{v - C_y}{f_y} \\ 1 \end{bmatrix} \tag{9}$$

4.4.3 Gimbal Rotational Transformation

Following the computed Camera Ray Vector that incorporates the updated intrinsic properties of the camera for both the original and cropped images, to translate these rays into real world coordinates, we need to estimate the rotational shift of the target location relative to the global coordinate system and subsequently project the rotated ray vector towards the ground intersection and finally the scale transformed vector into a real-world coordinate system (e.g. WGS-84 [102]). This interpretation is visually illustrated step by step in Figure 23.

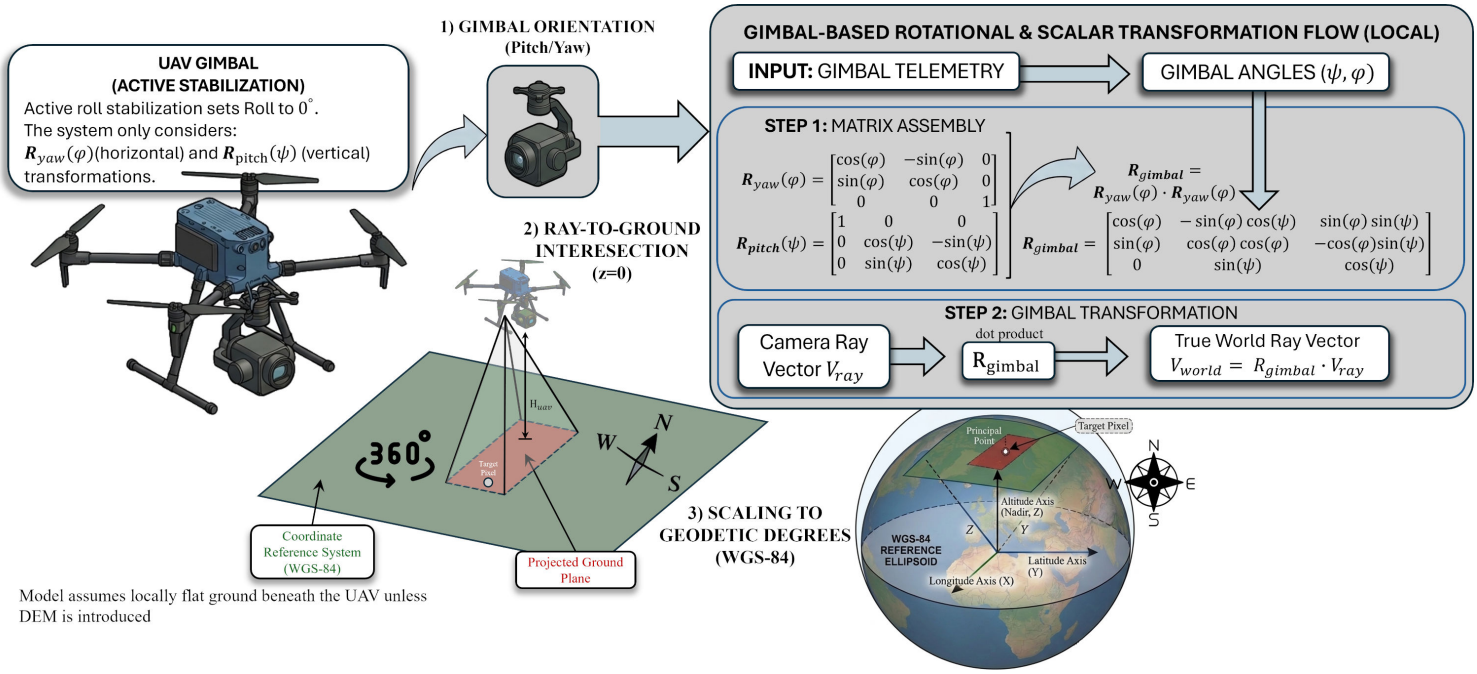


Figure 23: (1) Ray Vector Rotational Transformation, (2) Scaling to Ground Intersection and (3) Projection to Coordinate System WGS-84 [102]

Prior to projection, initially, a gimbal rotational transformation is employed using the extrinsic parameters provided by the 3-dimensional axis of the drone's gimbal across each frame. Notably, M30T's active stabilization mechanism, enables the roll extrinsic parameter to always be treated as zero, allowing the transformation to be modeled exclusively through **pitch** ψ and **yaw** ϕ telemetry angles. The simplified tracking rotation matrix $\mathbf{R}_{\text{gimbal}} \in \mathbb{R}^{3\times3}$ is constructed as a sequential transformation mapping pitch, controlling vertical gimbal tilt and yaw, controlling horizontal gimbal direction:

$$\mathbf{R}_{\text{gimbal}} = \mathbf{R}_{\text{yaw}}(\phi) \cdot \mathbf{R}_{\text{pitch}}(\psi) \implies$$

$$\mathbf{R}_{\text{gimbal}} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) & 0 \\ \sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & -\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} = \begin{bmatrix} \cos(\phi) & -\sin(\phi)\cos(\psi) & \sin(\phi)\sin(\psi) \\ \sin(\phi) & \cos(\phi)\cos(\psi) & -\cos(\phi)\sin(\psi) \\ 0 & \sin(\psi) & \cos(\psi) \end{bmatrix} \tag{10}$$

Finally, the dot product of the Rotational Matrix of the gimbal ($\mathbf{R}_{\text{gimbal}}$) with the Camera Ray Vector ($\mathbf{V}_{\mathbf{ray}}$) will produce the true World Ray Vector ($\mathbf{V}_{\text{world}}$):

$$\mathbf{V}_{\text{world}} = \mathbf{R}_{\text{gimbal}} \cdot \mathbf{v}_{\text{ray}} = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

4.4.4 Ray To Ground Intersection

Because a single 2D image lacks structural depth information, the physical distance along the world ray vector must be derived using the drone's relative altitude (H) above the ground

plane. Assuming a locally flat ground plane directly beneath the UAV, the distance scalar (s) from the camera lens to the ground target is calculated by projecting a ray to the ground level ($z = 0$):

$$s = \frac{-H_{uav}}{V_z} \tag{11}$$

Where V_z is the vertical component of the World Ray Vector (V_{world}), assuming V_z were equal to 1, it would mean the camera ray is pointing perfectly straight down at a 90-degree angle, directly beneath the drone, also known as Nadir viewing.

4.4.5 Solving the Depth Problem

In real-world applications, most landscapes may exhibit dynamic or hyperbolic topographic variance. Assuming a flat surface in our calculations, it may introduce severe displacement errors for given landscapes. One method to alleviate this is to point the gimbal directly towards the ground in Nadir viewing, thus, minimizing this effect of object displacement due to perspective. Another solution is to incorporate a dual-model methodology by leveraging Digital Elevation Models (DEM). The DEM acts as a discrete 2D depth plane mapping geographic coordinates to structural ellipsoidal orthometric elevations (Z_{DEM}).

$$s = \frac{Z_{\text{DEM}}(\phi_i, \lambda_i) - H_{\text{uav}}}{V_z} \tag{12}$$

For the given framework, we assume all surfaces are flat during our calculations, so we leverage Equation 11. Once the true distance scalar is computed, we can multiply it with V_{world} to scale the 2D plane in meters along the local horizontal tangent plane (East-North-Up frame) relative to the drone's position:

$$\begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = s \begin{bmatrix} V_x \\ V_y \end{bmatrix} \tag{13}$$

Here, Δx represents the metric distance to the target along the East axis, and Δy represents the metric distance along the North axis.

4.4.6 Scaling Meters to Geodetic Degrees

To convert these scalar metric offsets into fractional degrees of latitude and longitude, we leverage the anchor coordinate of the UAV that was mentioned previously (GPS_{uav}). Mathematically, this conversion translates the metric displacements ($\Delta x, \Delta y$) into angular changes based on the geometry of the Earth at the localized scene. To maintain structural simplicity within the framework, the system models the Earth using localized structural constants for the radius (R_M) and the prime vertical radius (R_N):

$$R_M = 6378137.0 \text{ m}; R_N = 6356752.3 \text{ m} \tag{14}$$

Using these geometric scaling constraints, the true horizontal metric ground displacements in meters is mapped into angular degree variations. The final true geographical coordinate parameter vector of the target object ($\mathbf{GPS}_{\text{tgt}}$) is formulated by adding these fractional degree offsets directly to the initial UAV reference anchor coordinates:

$$\mathbf{GPS}\text{tgt} = \begin{bmatrix} \phi_{\text{tgt}} & \lambda_{\text{tgt}} \end{bmatrix} = \begin{bmatrix} \phi_{uav} + \frac{\Delta Y}{R_M} & \lambda_{uav} + \frac{\Delta X}{R_N \cos(\phi_{uav})} \end{bmatrix} \tag{15}$$

This completes the geolocation pipeline, where 2D image pixels are projected into real-world Global Coordinates by passing a local pixel coordinate pair (u, v) from the UAV frame through this sequential four-step framework. The framework effectively computes displacements relative to the principal point of the frame, applies 3D gimbal transformations, scales the line-of-sight vector using relative UAV altitude, and projects the final target position into a standardized, geodetically accurate WGS-84 position vector.

Chapter 5: Experimental Geolocation Applications and Advanced Mapping Techniques

Building upon the detected objects, this chapter expands on two real-world applications fully automated using computer vision along with the geolocation estimation pipeline established from the preceding chapters. By combining edge-based optimized object detection and deep semantic segmentation with rigorous geodetic projection mechanisms, the pipeline is evaluated across two experiments with great importance in aerial mapping and target detection. To adopt the mathematical methodology of the geolocation pipeline thoroughly described in Chapter 4 into real-world concrete operational applications, it is fundamental to adopt vegetation topographical mapping methods. The first application addresses micro-scale environmental monitoring through a **Probabilistic Multi-source Geolocation Estimation** (PMV-Geo) of Semantically Segmented Salt Marsh canopies. This application aims to perform UAV-based fine-grained vegetation mapping techniques entirely through a RGB-based camera from multiple UAV images. Salt marshes are highly sensitive and among the most important vegetation types supporting climate conditions on a local and global scale, they effectively control the hydrodynamics of lagoons, thus, reducing inundation and flooding events and they supporty neighboring ecological systems by providing rich nutrients for the aquatic and plant life. The second application addresses standard surveillance techniques by integrating optimized YOLO models for detection and tracking of vehicles and pedestrians in UAV-based footage, coupled with the geolocation pipeline to produce their subsequent GPS coordinates. This methodology, operates as a baseline implementation and to be adopted into real-time UAV-based applications.

For both applications separate strategic evaluation techniques were adopted to identify the limits and precision of the geolocation estimation pipeline. For the first application, PMV-Geo, a highly robust and advanced technique was adopted to evaluate the geolocated salt marsh vectors on ground-truth labeled data, namely, **Symmetric Cross-Correlation Evaluation** (SCCE). This method was adopted to combat the significant temporal difference of the ground-truth (April 6, 2024) and predicted salt marsh vectors (July 17, 2025). Due to the dynamic behaviour of salt marshes constantly expanding and eroding, 1:1 geospatial comparison would produce high geolocation errors up to $\pm 25\ m$. However, by isolating marsh bodies that have not shifted spatially throughout the study area, geolocation accuracy can effectively be estimated without the harsh salt marsh temporal shifts. The second application performs standard point-to-point evaluations using the Circular Error Probable (CEP) metric to monitor errors and based on the results perform optimizations to improve geolocation accuracy.

5.1 Application I: Probabilistic Multi-View Geolocation of Semantically Segmented Salt Marshes

This application operates by integrating Semantic Segmentation models trained under a custom marsh canopy dataset to produce geolocated semantic masks for each image through the four-step mathematical geolocation pipeline. It resolves a major issue of the pipeline where it produces single-frame semantic GPS polylines. Rather than producing single-frame coordinate vectors, the system treats each projected pixel as a localized spatial probability distribution through a multi-source merging technology, Figure 24. This innovative application produces ultra-high-resolution geographical marsh vectors that satellite imagery fails to detect due to ground sample distance limitations. It enables ecologists and researchers to utilize it as a tool to

produce ultra-high-resolution spatial maps as ground-truth data to train state-of-the-art semantic segmentation models for marsh cover type detection through satellite imagery. Conversely, it can be utilized to detect with ultra high accuracy potential annual marsh morphodynamic shifts through erosion and dilation.

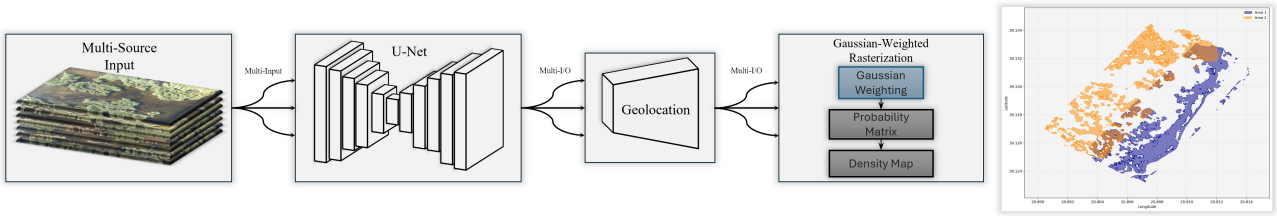


Figure 24: Probabilistic Multi-View Geolocation Application Design

The next section thoroughly expands on the ecological importance of salt marshes and their monitoring

5.1.1 Ecological Importance of Salt Marshes

Salt marshes represent one of the most ecologically significant vegetation ecosystems, occupying a large portions of coastal lagoons such as the Venice Lagoon and the Amvrakikos Gulf. These halophyte-dominated ecosystems are characterized by a unique ability of biogeochemical processes that enable them to store large Carbon dioxide (CO_2) and methane (CH_4) [105], effectively operating as natural filters for the environment. They are also known to play an important role in delivering substantial benefits to neighbouring vegetation and acquatic ecosystems through vertical sediment accretion primarily by acting as tidal barriers, effectively slowing current velocities. These halophyte-dominated communities function as a critical biogeochemical interface, mediating the flux of nutrients, sediments, and energy between land and sea. The ecological importance of salt marshes is multifaceted: they serve as prolific carbon sinks, sequestering organic carbon at rates up to ten times greater than other terrestrial vegetation types per unit area, a process referred to as “blue carbon” accumulation [106, 107]. Furthermore, salt marshes provide indispensable nursery habitats for numerous marine species, including commercially valuable fish and invertebrates, thereby sustaining coastal fisheries and biodiversity [108, 109].

Beyond their ecological roles, salt marshes deliver substantial ecosystem services of considerable economic value. These include natural water purification through the retention and transformation of pollutants (e.g., heavy metals, excess nutrients), shoreline stabilization through sediment accretion and wave energy dissipation, and flood mitigation for inland communities [110]. Despite their ecological value, coastal marshes are constantly threatened by various drivers such as urbanization, the development of anthropogenic infrastructures shifting tidal currents, sea level rise, extreme meteorological events [105] inundation event frequencies [111, 112]. These drivers can lead to significant marsh and tidal morphodynamic shifts, leading to severe marsh erosion, subsequently impacting surrounding ecosystems and conversely the global climate.

5.1.2 Monitoring of Coastal Salt Marshes

These challenges facilitates the need to monitor coastal salt marshes systematically to observe their long-term behaviour. Traditional surveying methodologies for salt marsh monitoring rely on labor-intensive field campaigns where ecologists physically walk through dense marsh landscapes with handheld GPS receivers to collect precise data. These manual efforts follow strict tracking guidelines, such as the National Oceanic and Atmospheric Administration's (NOAA) NERRS protocol [113]. While these on-site measurements yield high precision ground-truth data, they are severely limited by spatial coverage, high operational costs, and time inefficiencies. This makes them unfeasible for continuous, large-scale monitoring of expansive and difficult-to-access coastal zones. To overcome these limitations, remote sensing technologies have emerged as a transformative alternative. Satellite-based multispectral imagery (e.g., Sentinel-2, Landsat-8/9) has been extensively employed for broad-scale salt marsh mapping. Several studies have been conducted to assess the long-term dynamic evolution of salt marshes. Among these studies, remote sensing is most prominent for this domain by utilizing multi-spectral or hyper-spectral satellite imagery. To map coastal marshes, several studies have developed and employed models relying on methods such as Spectral Analysis [114], Machine Learning (ML) [115], Ensemble Learning (EL) [116], and Deep Learning (DL) [117]. Many of these methods heavily rely on annotation vectors overlaying marsh canopies to train such models. Such digital vectors are provided by professional ecologists requiring field-based GPS surveys that are often labor-intensive and challenging due to the vast scale of coastal lagoons and the difficulty in navigating them.

Conversely, commercial satellites such as Sentinel-2 and LandSat-8, while they are free for use, present limitations in resolution, with each satellite sensor supporting different Ground Sample Distances (GSD) per square pixel. Sentinel-2 offers a moderate GSD of up to 10 meters for its primary bands [118], while LandSat-8 provides a much lower resolution of 30 meters per square pixel GSD [119], resulting in the loss of granular spatial detail. While other commercial satellites from platforms like WorldView provide higher resolution imagery with some bands delivering a GSD as low as 31 centimeters per square pixel, acquiring such data is very expensive and not always feasible for large-scale long-term coastal monitoring.

5.1.3 Monitoring Coastal Marshes through UAV Imagery

This thesis developed a very practical solution that solves real-world problems such as the high cost of ultra-high-resolution satellite imagery and the labor-intensive need to perform field-based GPS surveys through the use of Multi-View Geolocation Estimation using UAV Photogrammetry. In this context, the integration of Unmanned Aerial Vehicle (UAV) combined with high-resolution RGB or multi-spectral sensors, coupled with automated semantic segmentation techniques, offers a promising paradigm for ultra-high-resolution salt marsh mapping [**?**]. Semantic segmentation models produce pixel-level classification masks that delineate vegetation types, bare sediment zones, and inundated regions with sub-meter precision. Deep convolutional neural networks (e.g., U-Net, DeepLabV3+, Mask R-CNN) and Vision Transformer architectures have demonstrated remarkable efficacy in extracting complex spectral-spatial features from aerial imagery, enabling robust discrimination of ecologically distinct classes even under challenging conditions such as variable illumination, tidal fluctuations, and mixed-pixel phenomena [120]. The application of these advanced CV techniques to UAV-derived orthomosaics facilitates the generation of detailed, geolocated marsh habitat maps that support longitudinal

ecological assessments, disturbance detection, and adaptive conservation strategies. To support the task of geolocating complex vegetation types such as coastal marshes, a large quantity of data needs to be acquired through planned UAV flights over a large coverage of the Amvrakikos Gulf.

5.1.4 Data Acquisition

Field surveys were executed on July 17, 2025, within the northern region of the Amvrakikos Gulf. This coastal ecosystem represents the largest and most ecologically complex wetland in western Greece. The flight plan lasted a total duration of 40 minutes and 54 seconds, spanning from 18:13:31 to 18:54:25 UTC, during which a total of 693 high-resolution images were captured utilizing the wide-angle sensor of the DJI M30T drone. The aspect ratio of the captured images is 4:3 (4000×3000), which defines the specific intrinsic parameters for our use-case.

Moreover, to minimize the geometric perspective distortion and alleviate target displacement errors caused by topographic variance, the gimbal was locked in a nadir viewing configuration with a pitch angle of $-89.90°$ pointing directly towards the ground. The survey was divided into two distinct flight tracks designed to map separate geographic sectors of the lagoon system:

- **Area 1:** Consists of 364 images captured over a 19-minute and 20-second window (18:35:05 to 18:54:25). The flight track covered a total linear distance of 7.53 km (7532.46 m) at a stable relative altitude mean of 113.03 m (ranging from a minimum of 112.77 m to a maximum of 113.30 m). Geographically, the surveyed area defines a spatial bounding box spanning approximately 910 m along the north-south latitudinal axis and 820 m along the east-west longitudinal axis.

- **Area 2:** Consists of 329 images captured over a 17-minute and 29-second window (18:13:31 to 18:31:00). The flight track traversed a total linear distance of 6.86 km (6856.65 m) at a mean relative altitude of 115.09 m (ranging from a minimum of 114.81 m to a maximum of 115.36 m). Geographically, the surveyed sector defines a spatial bounding box spanning approximately 900 m along the north-south latitudinal axis and 750 m along the east-west longitudinal axis.

The cumulative ground area covered across the entire survey evaluates to approximately 1.06 km^2 (105.52 hectares). This spatial coverage was derived by computing the mathematical union of the convex hull of projected image footprints across both flight paths, where Area 1 accounted for 0.54 km^2 (54.25 hectares) and Area 2 accounted for 0.51 km^2 (51.27 hectares). Individual image footprints were estimated by intersecting the sensor's optical ray bounds—defined by the hardware specifications of the DJI M30T wide camera ($\approx 73°$ horizontal field of view and $\approx 57°$ vertical field of view)—with the instantaneous relative altitude recorded in the XMP metadata of each frame. The spatial distribution of the captured points across the Amvrakikos Gulf environment is structurally illustrated in Figure 25.

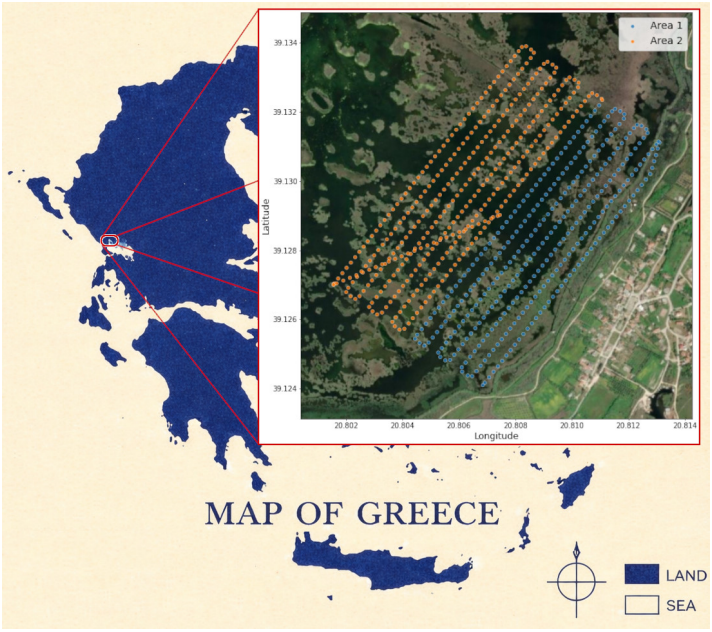


Figure 25: GPS Points of the two distinct Areas

To establish the physical ground resolution across the surveyed areas, the Ground Sample Distance (GSD) must be calculated. Given a stable operational flight altitude (H) of approximately 115 m relative to the takeoff point and utilizing the hardware dimensions of the DJI M30T wide camera sensor, comprising of a physical sensor width ($S_w = 6.4$ mm), a physical sensor height ($S_h = 4.8$ mm), an optical focal length (f) of 4.5 mm, and native dimensions of 4000×3000 px, the physical GSD per square pixel is determined as follows:

$$GSD = \frac{S_w \cdot H}{f \cdot I_w} = \frac{6.4 \text{ mm} \cdot 115 \text{ m}}{4.5 \text{ mm} \cdot 4000 \text{ px}} \approx 4.09 \text{ cm/pixel} \tag{16}$$

This sub-decimeter GSD, establishes that each individual pixel maps a precise 4.09 cm $\times$ 4.09 cm physical footprint on the ground plane, resulting in the ultra-high-resolution spatial vector to capture fine-grained vegetation features and subtle margins with extreme precision.

5.1.5 Computer Vision Pipeline Training & Evaluation Dataset

Transforming the raw aerial imagery into a structurally sound dataset fit for deep learning required a rigorous curation, semantic annotation, and data augmentation pipeline. For this study, the 364 images comprising Area 1 were selected as the primary supervised baseline. Using a cloud-based pixel-level labeling suite, manual semantic segmentation masks were generated for every frame in this set. The primary target class was defined strictly as active salt marsh canopy (halophyte vegetation matrices), while all other land cover features such as open water channels, barren mudflats, structural objects, and distinct non-marsh terrestrial vegetation were grouped together and assigned to a single background class. Conversely, the 329 images from Area 2 were intentionally reserved as an unlabeled dataset to accommodate time constraints, providing an independent, raw sequence for multi-frame qualitative model validation and spatial inference testing. To avoid overfitting and improve generalization of the segmentation networks under varying spatial orientations, spatial data augmentations were applied to the annotated training array. These consisted of stochastic horizontal and vertical flipping

transformations implemented with a execution probability of $p = 0.5$, forcing the models to learn invariant structural descriptors of marsh boundaries independent of flight direction. Following augmentation, the curated Area 1 dataset was partitioned into isolated subsets using a standard 60/20/20 split for training, validation, and testing profiles, respectively. The training subset provides the primary gradient updates for optimization, the validation subset provides unbiased evaluation for hyperparameter tuning and model checkpointing, and the final test subset remains completely held out to verify the definitive pixel-level segmentation accuracy of the architectures prior to feeding the output boundary masks into the geolocation estimation pipeline.

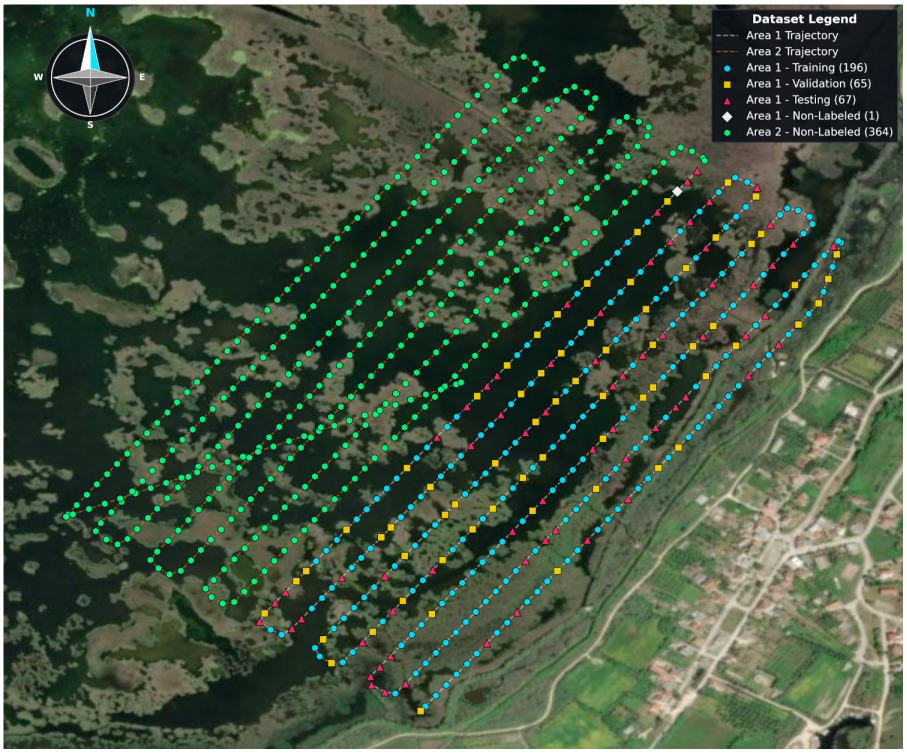


Figure 26: Visual Illustration of the Dataset Distribution

5.1.6 Geolocation Pipeline Evaluation Dataset

To evaluate PMV-Geo's precision, a Esri World Imagery satellite basemap was retrieved for the corresponding location and was stored as a '*.tif*' file while sustaining geospatial data. The image was labeled using SAM2 [121] (Segment Anything version 2). SAM2 operates as a real-time zero-shot semantic segmentation tool to generate semantic masks using promptable visual segmentation in both images and videos. To annotate the geospatial satellite image with masks overlaying salt marsh pixels, a tool was developed to annotate it using positive and negative points. Image tiling was adopted along with SAM2-hiera-large to increase the precision and compensate for the small spatial context provided by SAM2 relative to the high resolution of the geospatial image. Figure 27 illustrates the annotated satellite basemap along with the marsh mask extracted using tiled SAM2.

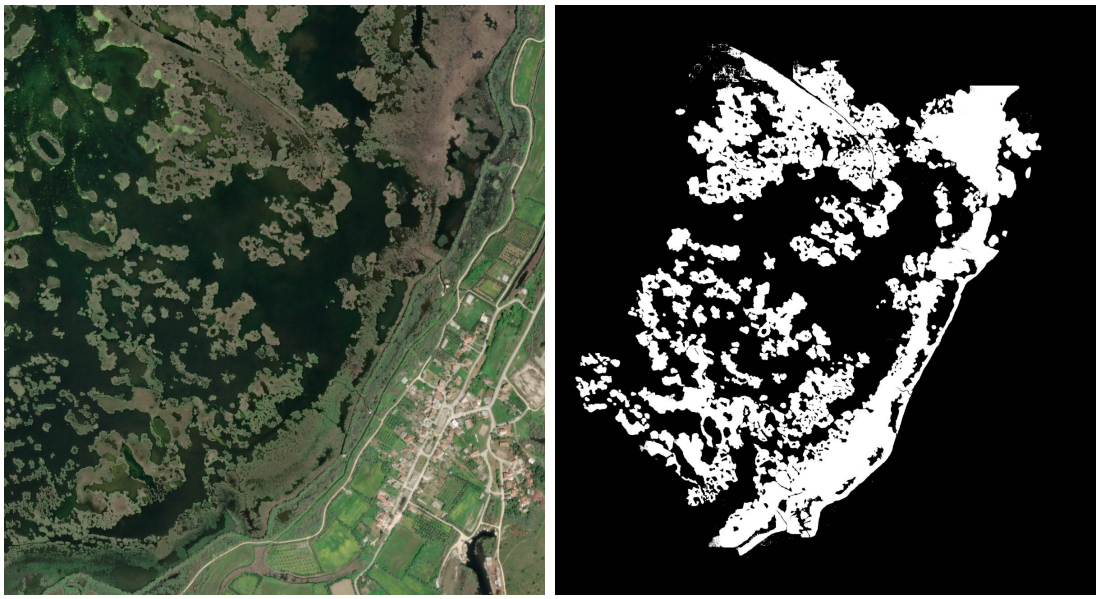


Figure 27: left: Esri World Imagery Basemap Acquired on April 6, 2024. right: Extracted semantic mask using SAM2 geospatially overlaying Marsh pixels

The annotation was intentionally developed to extend further than the UAV survey's footprint. Additional processing of the annotation mask includes clipping the mask strictly to the UAV survey's perimeters, thus, resulting in more sufficient evaluations.

5.1.7 Detection of Coastal Salt Marshes

The practical execution of the marsh boundary mapping pipeline relies on a unified deep learning framework to process the high-resolution aerial imagery captured during the Amvrakikos Gulf survey. To extract geospatial data, a conventional U-Net architecture was selected for semantic image segmentation due to its encoder-decoder configuration, which explicitly preserves low-level structural and edge context, that is critical for fine-grained vegetation mapping. Moreover, for additional evaluation, YOLO26x [75] was included within the experiments to compare U-Net with more advanced modern architectures. The core objective of this pipeline is to perform pixel-level classification, categorizing images into distinct marsh canopies versus background topographies.

Training Framework

To optimize the model's accuracy, the choice of spatial input scale was carefully evaluated against sensor flight constraints. Operating at a relative flight altitude of approximately 115 m above the takeoff point with the DJI Matrice 30T wide camera configuration (6.4 mm sensor width, 4.5 mm focal length, and 4000 × 3000 native image resolution), the baseline physical Ground Sample Distance (GSD) was mathematically determined to be approximately 4.09 cm/pixel. This ultra-high-resolution GSD will ensure that any minor vegetation fragmentations such as tight tidal flow channel paths, gaps and transitional boundaries will be identified by the computer vision pipeline. Empirical multi-scale resolution experiments were carried out, to observe the computational trade-offs between model precision and performance by evaluating three distinct spatial inputs (256×256, 512×512, and 1024×1024 pixels). U-Net

was trained using these distinct input sizes and subsequently evaluated along with YOLO26x. To systematically optimize the semantic segmentation pipeline, the training process was structured to balance immediate convergence and model generalization.

Training Configuration

For U-Net, AdamW was utilized for dynamic weight adjustment [122], paired with a Binary Cross-Entropy with Logits (BCEWithLogitsLoss) objective function [123]. U-Net, is highly susceptible to overfitting and unstable training, meaning it should be addressed carefully. To address the instability of U-Net, learning rate was reduced to $1e-5$. YOLO26x was trained using standard configurations (End-2-End), along with the modern SOTA hybrid MuSGD optimizer [75]. Both models were trained for 25 epochs.

Training & Evaluation Results

The complete optimization trajectory illustrated in Figure 28. The model checkpoints were monitored continuously, while storing the weights that yielded the highest evaluated F1-score per epoch. As demonstrated by the performance metrics, the models started converging early at approximately 20 epochs. Extending the training loop beyond 25 epochs would increase the risk of model overfit. Thus, training was terminated at 25 epochs as an effective early-stop as a precaution to preserve spatial generalization for unseen data.

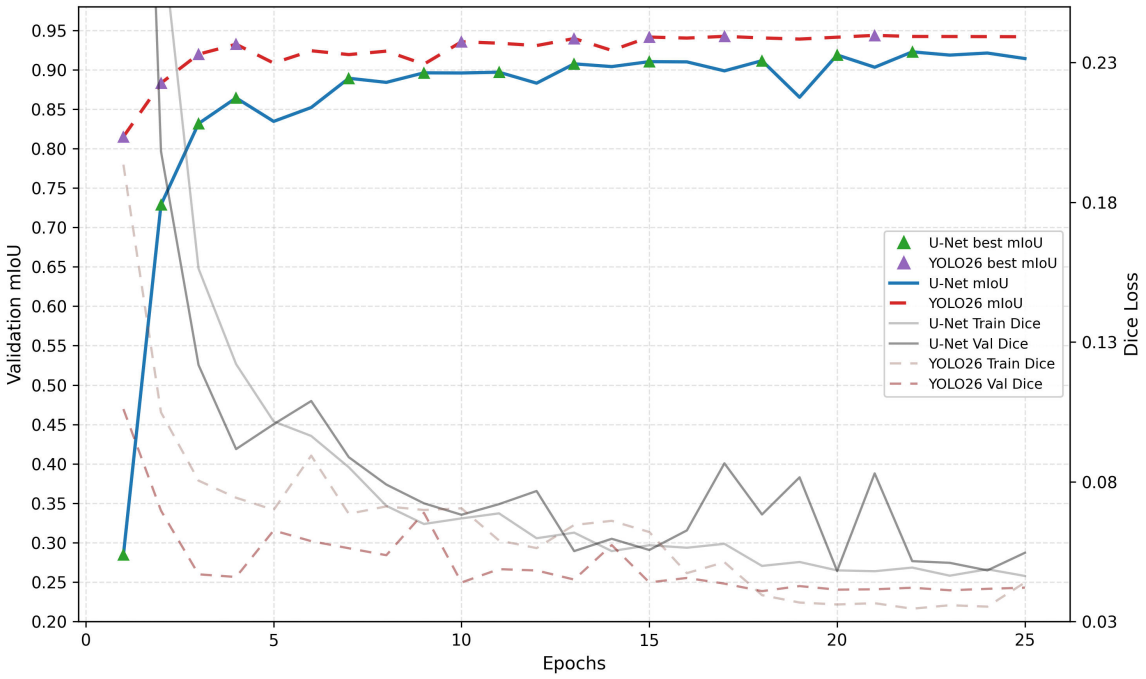


Figure 28: U-Net 1024^2 and YOLO26 ($imgsz = 1024^2$) Dice Losses and mIoU Performance Metric Scores

As expected, performance improved in proportion to the input size. U-Net ($imgsz = 1024^2$) achieved the highest mIoU score of 0.9227 and 0.9142 for the validation and training, respectively. The percentage increase between scales for $\times 512$ to $\times 1024$ was 1.66%, whereas for $\times 256$

a significant 7.17% mIoU increase. The increasing spatial context provided to U-Net significantly increased computational performance demand, reaching up to 772.73 Giga Flops. Most notably, within the experiments, YOLO26x-sem, drastically outperformed the conventional U-Net, which is traditionally known as a state-of-the-art pixel-level semantic segmentation model, developed in 2015, that was originally designed for Medical Imaging [124]. YOLO26x, outperformed U-Net across all metrics, achieving an all time high of 0.9436 mIoU score for the validation set while simultaneously being a significantly lighter model for inference. The results are further summarized in Table 9 reporting several metrics such as total mIoU for val/test sets, Recall, Precision, Accuracy and computational performance.

Model	**Batch Size**	**mIoU**$_{val}$	**Recall**	**Precision**	**Accuracy**	**mIoU**$_{test}$	**FPS**	**GFlops**
U-Net 256 × 256	32	0.8609	0.9341	0.9166	0.9441	0.8604	393.94	48.30
U-Net 512 × 512	16	0.9076	0.9636	0.9402	0.9664	0.9076	93.25	193.18
U-Net 1024 × 1024	4	0.9227	0.9684	0.9515	0.9706	0.9142	23.18	772.73
YOLO26x-sem	4	**0.9436**	-	-	**0.9729**	**0.9273**	**89.53**	**167.36**

Table 9: Semantic Segmentation performance comparison on the Area 1 coverage Dataset

During final model deployment, YOLO26x-sem was used to generate the inference predictions with the highest precision. The resulting semantic masks are upscaled to the native 4:3 frame dimensions using nearest-neighbor interpolation to preserve structural boundary edges. These finalized semantic boundaries are then passed into the Geolocation pipeline, subsequently projecting local predictions into three dimensional space and finally into a fully georeferenced WGS-84 vector. Figure 29 Illustrates the per-image semantic segmentation masks overlayed over the UAV imagery, in addition with the geolocated marsh canopies over a standard global coordinate basemap.

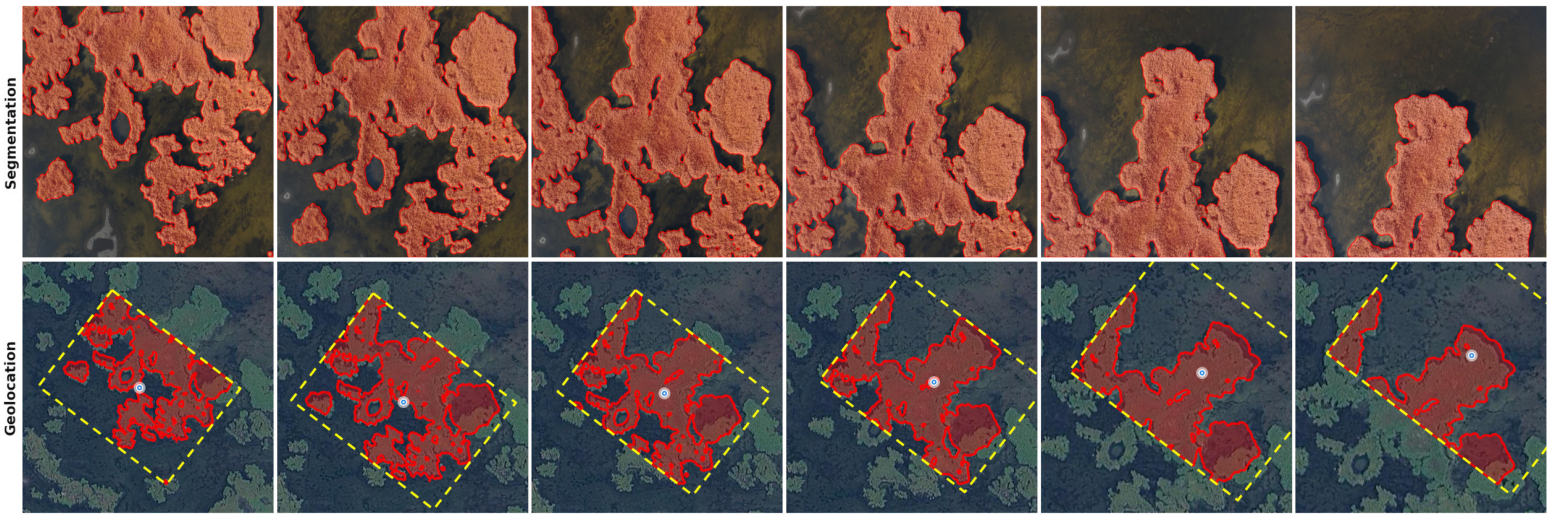


Figure 29: Geolocation Estimation Composite on the detection masks produced by YOLO26x. Basemap overlay captured on April 6, 2024, UAV imagery captured on July 17, 2025

The direct comparison (Figure 29) with the google maps basemap dating back to April 6, 2024 and the UAV Imagery retrieved in July 17, 2025, illustrates precise geolocation of the

marsh cover types. The geolocated masks reveals expansion and erosions, suggesting that newly developed marshes grew or were slowly eroded throughout that time period. These expansions and erosions are observed across several UAV images within the dataset. One Marsh expansion example within the dataset is visually illustrated in Figure 30.



Figure 30: UAV-derived salt marsh detection (July 2025) overlaying WorldView Imagery (April 2024) with dilated polylines indicating marsh expansion.

Moreover, these vectors can be utilized to compare any cover type erosion or expansion throughout that time period. While each of these vectors are produced each image individually, the next section expands on the methodology utilized to compute a unified global geospatial matrix overlaying geolocated marsh canopies.

5.1.8 Probabilistic Multi-View Geolocation using Gaussian-Weighted Rasterization

The computer vision pipeline initially is performed for individual frames, where the trained model performs dense, pixel-level semantic segmentation on each sequential aerial image. By passing these individual frame masks into the geolocation pipeline described in Chapter 4, the system projects the 2D matrix predictions into discrete geographic coordinate streams. However, while computing these geolocated coordinates for individual frames offers a rapid, automated solution for localized target extraction, it is inherently limited to per-image generation. Attempting to stitch or map independent frame predictions directly in the vector space results in severe geometric overlaps, spatial gaps, and fragmented polylines that lack topological clarity due to momentary telemetry noise, perspective changes and repeated inference results.

To resolve this limitation, individual frame inferences are aggregated into a singular, unified Global Grid defined within the WGS-84 geodetic coordinate system. The spatial boundaries of this global grid are calculated dynamically by extracting the minimum and maximum bounding coordinates from the complete drone flight telemetry data across the study area. The grid resolution is set to a fixed angular step of $\delta = 5 \times 10^{-6}$ degrees, which corresponds to an approximate spatial footprint of 50×50 cm per cell. This localized resolution provides sufficient spatial fidelity for geographic information systems (GIS) to map irregular marsh boundaries with sub-meter precision. To produce a spatial footprint in-line with the GSD (4×4 cm), the resolution of the global grid would be 12.5× larger than the 50×50 cm footprint resulting in

extremely dense polylines and extreme system memory consumption. Thus, for this application, the 50×50 cm footprint was used.

To effectively aggregate and project the prediction model's results onto the global grid without suffering from edge-alignment artifacts or duplicated polygons, the study area is initialized with two discrete tracking matrices: an accumulated probability matrix $\mathbf{P} \in \mathbb{R}^{M \times N}$ and a normalization density matrix $\mathbf{D} \in \mathbb{R}^{M \times N}$. The probability matrix accumulates the confidence scores of the semantic segmentation model weighted by a spatial reliability function, while the density map accumulates the cumulative optical weight distributed to each cell across all overlapping flight passes, effectively preventing data bias and structural fragmentation in high-overlap flight zones.

To populate these matrices, each pixel point $p(x, y)$ from an individual inference mask is transformed via the geodetic pipeline and mapped to its corresponding global grid cell index (i, j). A Gaussian-Weighted Rasterization strategy is employed to mitigate two primary geometric vulnerabilities inherent to monocular UAV photogrammetry: radial perspective displacement caused by topographic variance away from the nadir axis [2] and residual lens distortion, which is typically most pronounced near the sensor's physical boundaries [125].

This weighting mechanism prioritizes semantic predictions captured near the image principal point (c_x, c_y), where the optical ray path experiences the least spatial distortion and highest geometric fidelity. The reliability weight $W(x, y)$ assigned to a projected pixel is modeled using a bivariate isotropic Gaussian function centered at the image origin:

$$W(x, y) = \exp\left(-\frac{(x - c_x)^2 + (y - c_y)^2}{2\sigma^2}\right) \tag{17}$$

where σ represents the spatial decay variance parameter, adjusting how rapidly the weight devalues as a function of radial distance from the principal point.

The projection of each sequentially processed frame onto the global coordinate grid follows an iterative accumulator strategy. For every pixel prediction score $p_{k,x,y} \in [0, 1]$ from the upscaled semantic mask of frame k that projects into the grid cell at index (i, j), the probability and density matrices are updated concurrently:

$$\mathbf{P}_{i,j} \leftarrow \mathbf{P}_{i,j} + (p_{k,x,y} \cdot W_{x,y}) \tag{18}$$

$$\mathbf{D}_{i,j} \leftarrow \mathbf{D}_{i,j} + W_{x,y} \tag{19}$$

Once all 693 frames from the aerial survey have been rasterized and accumulated, the raw confidence scores are normalized by the cumulative spatial coverage density. The finalized, normalized global map matrix $\mathbf{M}_{i,j}$ is computed by performing an element-wise division:

$$\mathbf{M}_{i,j} = \frac{\mathbf{P}_{i,j}}{\mathbf{D}_{i,j} + \epsilon} \tag{20}$$

where ϵ is a small regularization constant introduced to prevent division-by-zero errors in grid cells that received no camera footprint coverage during the flight. Cells exceeding a specific probability threshold in $\mathbf{M}$ are then vectorized into continuous global polylines.

The projection of each sequentially processed frame onto the global coordinate grid follows an iterative accumulator strategy. For every pixel prediction score $p_{k,x,y} \in [0, 1]$ from the upscaled semantic mask of frame k that projects into the grid cell at index (i, j), the probability and density matrices are updated concurrently:

$$\mathbf{P}_{i,j} \leftarrow \mathbf{P}_{i,j} + (p_{k,x,y} \cdot W_{x,y}) \tag{21}$$

$$\mathbf{D}_{i,j} \leftarrow \mathbf{D}_{i,j} + W_{x,y} \tag{22}$$

Once all 693 frames from the aerial survey have been rasterized and accumulated, the raw confidence scores are normalized by the cumulative spatial coverage density. The finalized, normalized global map matrix $\mathbf{M}_{i,j}$ is computed by performing an element-wise division:

$$\mathbf{M}_{i,j} = \frac{\mathbf{P}_{i,j}}{\mathbf{D}_{i,j} + \epsilon} \tag{23}$$

where ϵ is a small regularization constant introduced to prevent division-by-zero errors in grid cells that received no camera footprint coverage during the flight. Cells exceeding a specific probability threshold in $\mathbf{M}$ are then vectorized into continuous global polylines.

By structuring the geolocation pipeline around this continuous raster-based aggregation framework, it systematically minimizes errors before generating any vector geometries. To formalize the specific computational role of each component within the global rasterization layer, the structural and geometric problems solved by each individual matrix are detailed below:

- **Geospatial Probability Map ($\mathbf{P}_{i,j}$):** Resolves the multi-view geometric contradiction problem. Because independent monocular frame projections generate isolated, frame-dependent boundaries drone movement and gimbal errors, this matrix spatially accumulates geographically similar pixels and weighs them in accordance to the Gaussian function. This mechanism effectively mitigates two key problems deviating from UAV Photogrammetry. The radial perspective displacement problem [2], where object elevation may discretely displace geographically mapped pixels and lens distortion [125], that is typically more profound at the frame's edges.

- **Geospatial Density Map ($\mathbf{D}_{i,j}$):** Effectively prevents overlapping geolocated estimations from adjacent UAV imagery by dynamically accumulating the total weights at each coordinate, thus reducing geolocation displacements or bias due to confidence spikes that would otherwise occur in zones with heavy overlap.

- **Finalized Geospatial Global Map ($\mathbf{M}_{i,j}$):** Resolves the structural fragmentation and topological ambiguity problem. By performing element-wise division of the accumulated probabilities by the total sensor footprint coverage density, it scales the raw inputs into a unified, normalized $[0, 1]$ confidence terrain. This allows the system to bypass the noisy, fragmented polyline intersections that occur when merging vectors directly, instead producing clean, continuous, and topologically consistent global vectors.

The visual illustration of this pipeline, Figure 31, details the spatial transformation from an individual Gaussian weighing function to a unified, density-normalized map representation.

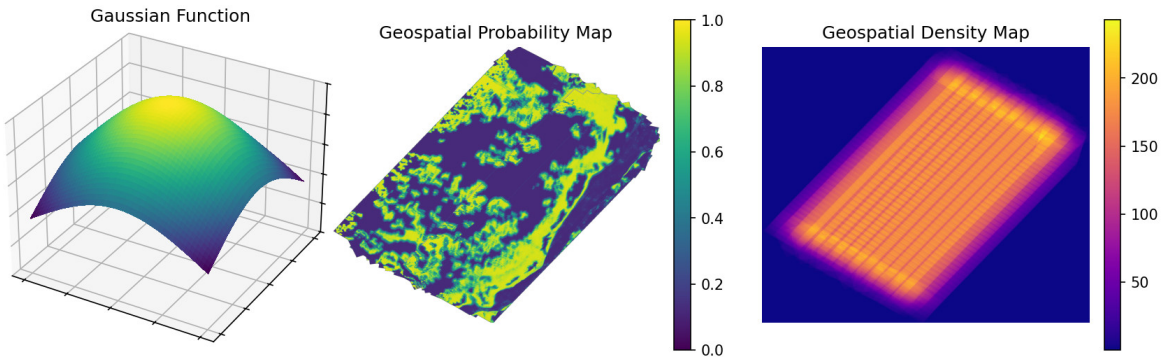


Figure 31: 3D Gaussian Function; Global Accumulative Probability Matrix; Global Density Map

Once the Probabilistic Multi-View Geolocation pipeline was applied to the whole set of marsh data a single Geospatial Global Map ($\mathbf{M}_{i,j}$) was produced for both Areas 1 and 2, the visual overlay of the whole marsh canopy scanned in July 17, 2025 can be conversely overlayed over a google satellite basemap as illustrated in Figure 32.

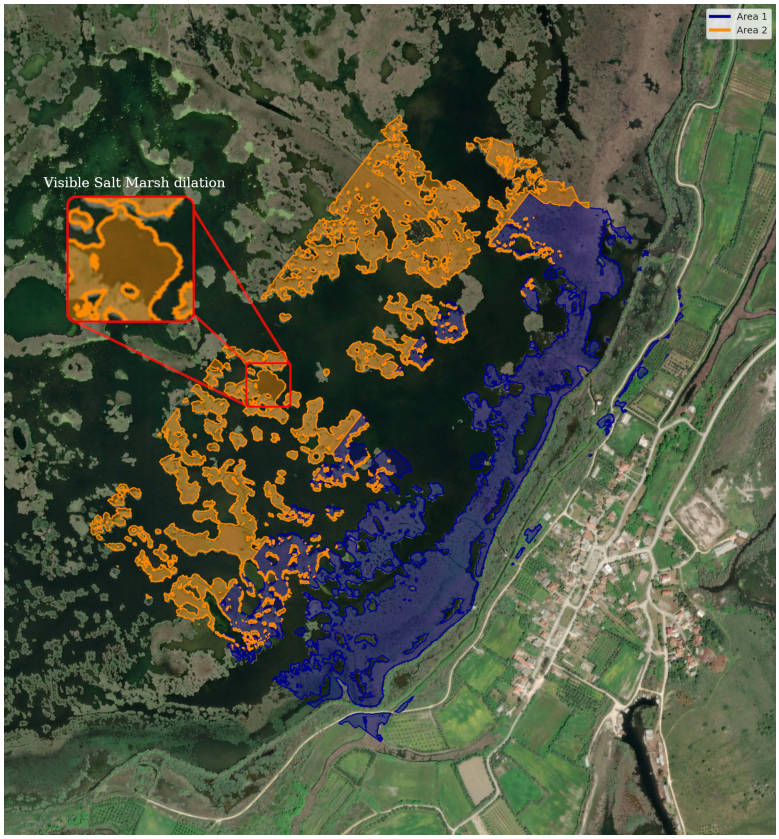


Figure 32: Unified Geolocated Salt Marshes of Area 1 and 2. Basemap Satellite image captured on April 6, 2024 [126].

The annotation mask described in a previous section retrieved using the Esri World Imagery basemap and the Geospatial marsh prediction produced by PMV-Geo are temporally disjoint, meaning, the satellite basemap was acquired on April 6, 2024, the UAV flight mission was conducted on July 17, 2025, 467 days apart. Salt marsh ecosystems are highly dynamic, within an annual gap, and they can drastically shift even within short time periods, making geolocation

evaluation all the more challenging. For simplification, the geospatial marsh vectors of year 2024 (Satellite annotation), will be addressed by T_0, year 2025 (UAV Survey) will be addressed by T_1.

5.1.9 PMV-Geo Evaluation Framework

Due to the dynamic behaviour of coastal salt marshes and temporal variation, a significant marsh expansion and erosion is observed across both segments T_0 and T_1, as visually illustrated in figure 34. Moreover, PMV-Geo's predictions having positionally drifted due to estimation error is an additional challenge that requires to be addressed to evaluate its precision. The positional drift of PMV-Geo along with the significant marsh erosion/expansion is vastly different symmetrically. PMV-Geo effectively captures the symmetry with absolute precision and is only limited by the computer vision pipeline accuracy, however, marsh expansion and erosion is often non-asymmetric and highly dynamic. This can be leveraged to evaluate PMV-Geo's positional drift error, without evaluating marsh areas that have expanded or eroded. Thus, a robust high-level evaluation framework was developed for this purpose to evaluate the geospatial precision of PMV-Geo, by filtering symmetrically proportional and disproportional geometries. To address these spatial asymmetries and evaluate the geospatial precision of PMV-Geo, the Symmetric Cross-Correlation Evaluation (SCCE) algorithm was developed.

5.1.10 Symmetric Cross-Correlation Evaluation (SCCE)

The SCCE algorithm mathematically formulates both T_0 and T_1 vectors by separately projecting them onto a shared high-resolution grid using the Universal Transverse Mercator (UTM) coordinate reference system. For our purpose, the shared grid was initialized with a 10 cm spatial resolution per pixel. The resulting grid size was produced to be approximately 13201x14048 pixels (width/height) at $0.1m/px$. Each vector boundary is decomposed into a set of localized spatial segments or windows, denoted as $S_0 = \{s_{0,1}, s_{0,2}, \ldots, s_{0,n}\}$ for T_0 and $S_1 = \{s_{1,1}, s_{1,2}, \ldots, s_{1,m}\}$ for T_1. Each sub-segment representing the ground truth (T_0) and PMV-Geo predicted (T_1) vectors contains a binary mask of the localized marsh, where a positive value (1), represents the solid marsh and a zero value water/non-salt marsh. For every possible sub-segment pair ($s_{0,i}$, $s_{1,j}$) (3468 sub-segments), a search strategy using the sliding-window method [127] is performed around the proximity of T_0 sub-segment across a broader T_1 search region. The overlapping geometries of $s_{0,i}$ and $s_{1,j}$ are bidirectionally structurally compared and paired based on a symmetry correlation score to determine how symmetrically similar both geometries are, while ignoring positional displacement and scale variation. All possible sub-segment pairs are compared between the proximity of $s_{0,i}$ and a larger search region of $s_{1,j}$ bidirectionally, and those exhibiting the highest mutual symmetry are paired and used for subsequent analysis. The resulting matches are ranked in descending order based on their correlation coefficients. By ordering each sub-segment pair from highest to lowest symmetry, SCCE effectively isolates structurally stable marsh shoreline features from areas impacted by severe marsh expansion/erosion, thus, enabling a stratified geospatial precision analysis. This is done to effectively mitigate PMV-Geo's estimations that may have positionally drifted due to algorithm error. SCCE was adopted to be fully modular, allowing for several advanced symmetry correlation systems to be adopted and utilized. While metrics like the Jaccard Index [64] and Dice Score Coefficients can be integrated to measure volumetric overlap and structural symmetry, both metrics do not account for the positional displacement drift which we require to evaluate the

precision of PMV-Geo. Thus, more robust symmetry correlation methods should be adopted for this purpose.

5.1.11 Structural Symmetry Scoring algorithms

Following the structural symmetry scoring algorithm, several techniques can be adopted and compared based on the current literature, the next part of this section will expand on several scoring techniques and compare it directly with its application to SCCE.

1. **Zero-Mean Normalized Cross-Correlation (ZNCC) on Solid Masks** [128, 129]
    - **Mechanism:** The sub-segments are converted from polylines into solid binary area masks (1 for marsh, 0 for water) and computes their spatial and structural similarity using a Normalized Cross-Correlation Coefficient [130].
    - **Application in SCCE:** Provides smooth, sub-pixel structural alignment by scoring the area-based spatial footprint, effectively evaluating geometry occlusion where PMV-Geo may have produced slightly larger shapes.
2. **Fast Directional Chamfer Matching (FDCM) on Area-Distance Fields** [131]
    - **Mechanism:** For each sub-segment a continuous Euclidean Distance Transform (EDT) of the target boundary $D_1(x, y)$ is produced and compared with a solid area mask M_0.
    - **Application in SCCE:** Penalizes spatial/structural mismatches along tidal channels by accumulating distance errors across the entire predicted marsh area. This ensures that the predicted geometry fits the actual target boundary rather than matching geometries proximity-wise.
3. **Modified Hausdorff Distance (MHD) on Dual Area-Distance Fields** [132]
    - **Mechanism:** Computes a symmetric morphological penalty by taking the maximum of two directional averages: the average distance from the T_0 solid area to the T_1 boundary, and the average distance from the T_1 solid area to the T_0 boundary.
    - **Application in SCCE:** Replaces outlier-sensitive extreme point-to-point distances with an averaged bidirectional structural metric, yielding a stable score of morphological fidelity that explicitly captures both severe marsh expansion and erosion.

Following the structural matching of each pair, SCCE calculates the physical spatial offset between the exact geographic centers of $s_{0,i}$ $s_{1,i}$ sub-segments, thus, estimating a straight-line displacement for both pairs, while tracking the latitudinal and longitudinal offsets and remaining immune to minor localized deformations along vector edges. Finally, by continuously computing the most optimal matches along the entire coastline, SCCE produces a displacement vector paired with correlation confidence scores. By stratifying the sub-segment pairs based on these correlation coefficients, the algorithm effectively isolates structurally stable marsh shorelines (high correlation) from areas impacted by severe ecological deformation (low correlation). Therefore, by evaluating high correlation pairs, we will ensure a more granular and rigorous structural analysis where natural marsh expansion or erosion does not penalize the overall geospatial precision metrics. Figure 33, thoroughly illustrates a geospatial heatmap of every ranked sub-segment pair produced using T_0 and T_1 geospatial vectors. Each sub-segment pair

mark is color-coded based on their corresponding symmetry score as seen in the colorbar. Moreover, the plot visualizes three sub-segment pairs based on their score percentiles, from Highest Correlation **A.** (Rank 1/3684), Median Correlation **B.** (1734/3468) and lowest correlation **C.** (Rank 3468/3468)

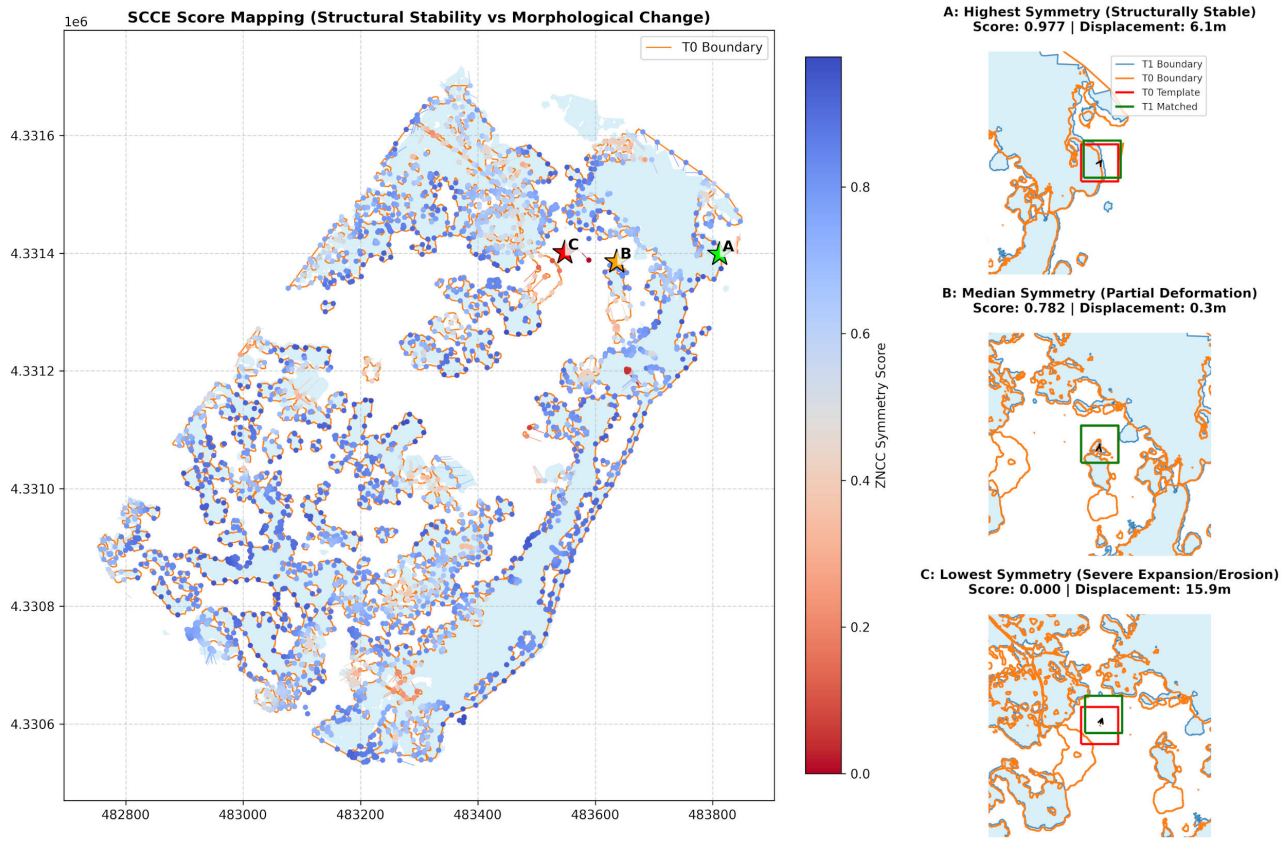


Figure 33: **Left**) SCCE Geospatial Score Heatmap where the orange outlines lines represent the T_0 marsh vector and blue overlay T_1 marsh vector. **Right**) Correlation Symmetry visualizations of top, median and lowest percentile.

5.1.12 Isolating Geolocation Error to Evaluate Symmetry Scoring Techniques

While SCCE can effectively evaluate the geospatial precision of PMV-Geo, it is important to consider isolating the efficiency of the symmetry correlation techniques and identify whether they play a critical role in the evaluation of PMV-Geo. Since the T_1 vector naturally does not match T_0 shape wise, a definitive method to evaluate the true performance of each symmetry correlation algorithm is to rely on a synthetically modified T_0 vector. This perturbation-based technique was adopted to isolate model error resulted by PMV-Geo and rely primarily on the ground-truth vector (T_0) and a synthetic perturbed T_0 vector. To evaluate and quantify each technique, two perturbation scenarios are adopted using a perturbed synthetic copy of the T_0 vector: 1) translation of the vector ($dx = 15.0m$ and $dy = -10.0$). 2) translation + morphological erosion/expansion of randomly selected 30% sub-segments. Each technique is quantified using mean displacement error, RMSE and 95th percentile error for both perturbation scenarios.

5.1.13 Results

As mentioned in preceding sections, the two temporally variant vectors (T_0, T_1) are geometrically vastly different across several locations, however, a large marsh portion has remained geometrically similar across both temporal points. Direct comparison of the two geospatial marsh vectors (T_0 and T_1), reveals these behaviours as illustrated in Figure 34.

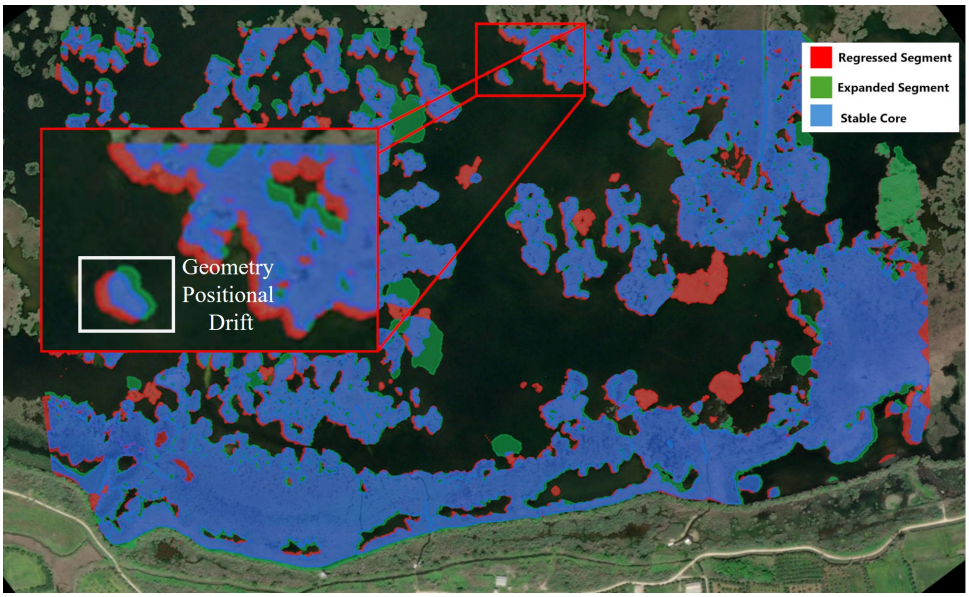


Figure 34: Temporally Disjointed Marsh Vector Segments

The blue overlay represents the **stable core**: The marsh geospatial points overlapping each other in both T_0 and T_1. The red overlay represents the **regressed segment**: Any marsh in T_1 that has eroded since T_0. The green overlay represents the **expanded segment**: Any marsh in T_1 that has expanded since T_0. The vector comparison reveals how in several geometries and locations, geospatial displacement was especially more prominent even though the geometries were structurally similar. To quantify and evaluate the geospatial drift of both marsh vectors through SCCE, the centroids of each sub-segment pair was compared and subsequently evaluated using the conventional Haversine Formula [133].

Symmetry Scoring Technique Evaluation

Before quantifying and evaluating the geospatial drift of PMV-Geo through SCCE, the best performing symmetry scoring technique needs to be identified. Each symmetry scoring technique was compared with two perturbation scenarios, as described in Section 5.1.12, the validation results revealed interesting distinctions between each algorithm. In the first scenario, ZNCC performed adequately, achieving an all time low of 0.43 mean displacement error, however, RMSE, peaked at 4.12 m indicating that a handful of sub-segments pairs were miss-matched as complete outliers, thus, heavily penalizing the error metric. Whereas MHD, performed the best and remained stable across all metrics, mean error, RMSE and 95th percentile error, conversely, FDCM had the worst performance and struggled to symmetrically match and correlate sub-segments. For the second scenario, the same was applied across all techniques, where MHD had the best performance and FDCM performed the least.

Augmentation	Method	Mean Error (m)	RMSE (m)	95th Pct Error (m)
T_0 + **Translation**	**ZNCC**	0.43	4.12	0.12
	FDCM	2.77	11.20	35.63
	MHD	0.08	0.08	0.12
T_0 + **Translation** + **30% Morph. Deformation**	**ZNCC**	23.30	26.84	44.37
	FDCM	25.20	28.99	52.82
	MHD	22.82	25.85	42.62

Table 10: Quantitative Displacement Error Comparison under Synthetic Augmentations

Quantitative results limit the scope of analysis, thus, to identify discrete algorithmic errors and behaviours visible within the synthetic dataset, Figure 35, illustrates a Complementary Cumulative Distribution Function (CCDF) of the displacement error (m). CCDF plots the probability where the displacement error exceeds a given threshold ($P(X > x)$). The best performing algorithm will exhibit a sharper slope, closer to the bottom-left axis, indicating a lower probability of large displacement errors. For Scenario 1), both ZNCC and MHD perform with near pixel-perfect performance, whereas for FDCM, it failed in the lower end probabilities. This an ideal and expected outcome where two symmetry correlation algorithms perform perfectly without geometry deformation. For Scenario 2), ZNCC initially shows virtually zero error for the top 5% to 8% of the distribution and decays much lower than MHD, however, on the upper middle half MHD catches up to ZNCC and performing slightly better than ZNCC in several occasions across the plot.

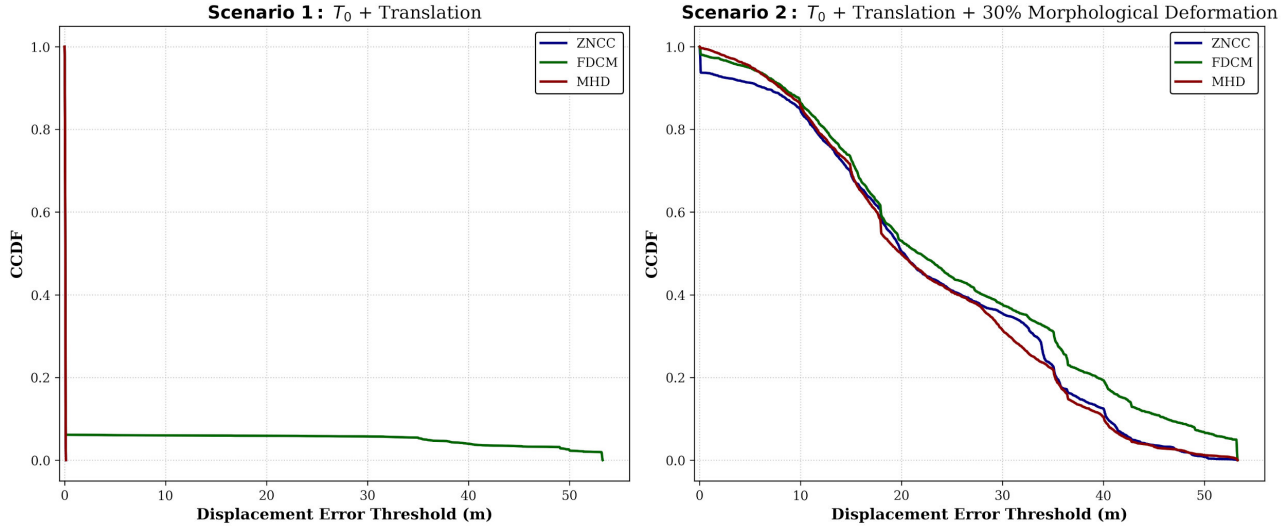


Figure 35: CCDF Probability Distribution of Displacement Error for Scenarios 1) & 2) under T_0 and Synthetic T_0 datasets.

While synthetic datasets may produce a baseline to validate and identify an optimal symmetry correlation algorithm (ZNCC, MHD, FDCM), they fundamentally fail to capture natural homogeneous dynamic erosion/expansions of coastal marshes. In Scenario 2, Table 10, introducing a randomly selected 30% morphological deformation resulted in a complete collapse across all metrics where MHD appears to perform marginally better than the other algorithms. In contrast, natural marsh erosion and expansion is often localized and highly homogeneous, making them difficult to artificially simulate. Moreover, following the next evaluation of the symmetry scoring techniques, it is proprietary to evaluate the PMV-Geo generated data that facilitates natural marsh erosion and expansion. The evaluation of each symmetry correlation

algorithm under T_0 and T_1 datasets (ground-truth and PMV-Geo, respectively), highlight entirely different results with ZNCC performing vastly better with a uniformly lower probability of failure across all displacement errors than MHD and FDCM significantly underperforming with a higher probability of failure, indicating it's FDCM is extremely sensitive to outliers, which is also addressed by MHD [132].

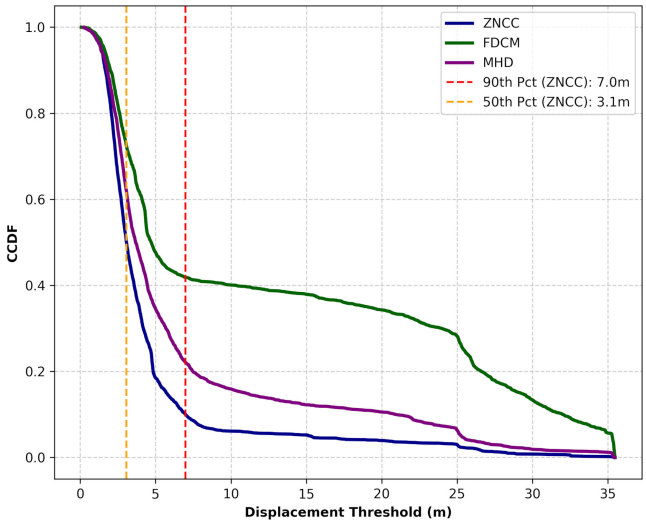


Figure 36: CCDF Probability Distribution of Displacement Error per Symmetry Correlation algorithm under T_0 and T_1 datasets.

Overall, the best performing symmetry scoring technique is ZNCC, due to its robustness in the natural dynamics of coastal salt marshes. By determining the best algorithm for our use-case, the next step is to perform a full quantitative and qualitative analysis of the geospatial precision of PMV-Geo and determine its true precision.

Evaluating PMV-Geo Geospatial Precision

Initial evaluations indicate that the highest concentration of ZNCC scores fell within the 0.7 to 0.95 percentile range. This suggests that a significant portion of the marsh vectors remained structurally stable between temporal points T_0 and T_1, as illustrated in Figure 37.

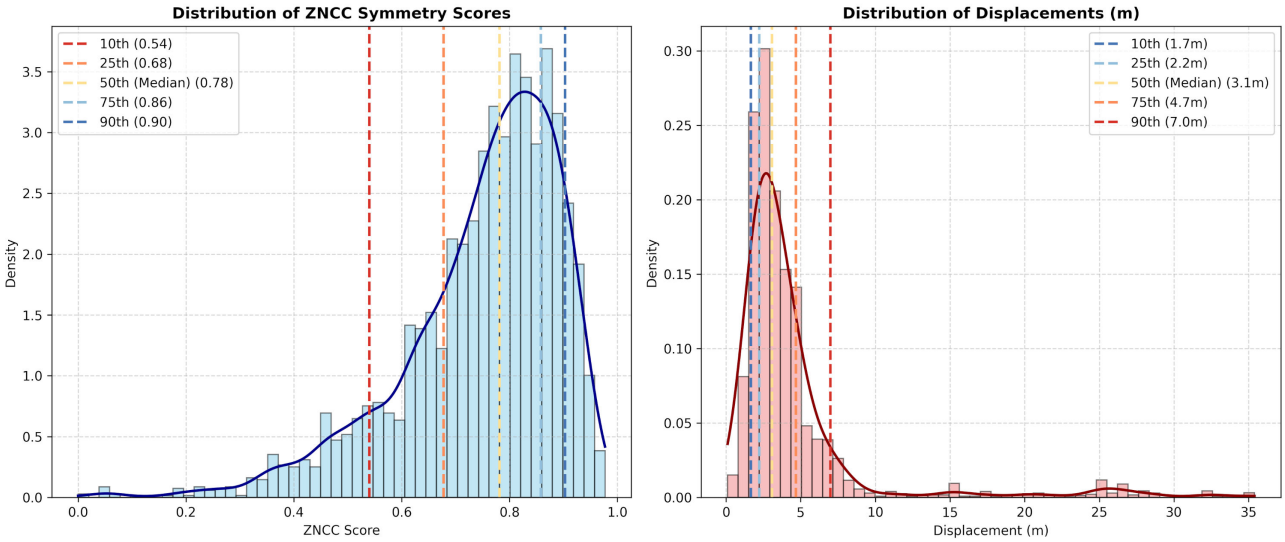


Figure 37: SCCE Distribution Of ZNCC Symmetry Scores and Pair Displacement in meters

To further quantify the percentiles and evaluate the true geospatial precision of PMV-Geo, the ranked pairs were stratified into percentile categories (<60, 60-85, 85-90, 90-95 and 95>, including overall) where higher percentile categories represent how symmetrically similar a pair is and was ranked accordingly. In proceeding evaluations, each percentile category reports N pairs, mean displacement (m) and root mean square error (RMSE) (m). Table 11, reports these evaluations for each category particularly using the ZNCC on distance transforms, symmetry scoring technique. Based on the quantified results, we can conclude that the true overall geolocation estimation error is approximately 2.9 to 3.5 meters.

Category	**Percentile** (%)	N **Pairs**	**Mean Disp. (m)**	**RMSE (m)**
Exceptional Correlation	≥ 0.95	45	2.90	3.16
Very High Correlation	$0.90 - 0.95$	333	3.15	3.42
High Correlation	$0.85 - 0.90$	564	3.17	3.47
Moderate Correlation	$0.60 - 0.85$	2025	4.05	5.58
Low Correlation	< 0.60	501	9.07	13.34
Overall	**All Scores**	**3468**	**4.53**	**6.86**

Table 11: Correlation Percentile Thresholds and Corresponding Displacement Metrics

5.1.14 Application I: Conclusion

The integration of Unmanned Aerial Vehicle (UAV) photogrammetry with deep semantic segmentation architectures provides a robust, scalable alternative to traditional, labor-intensive field-based GPS surveys and lower-resolution satellite monitoring. By leveraging high-resolution RGB imagery captured by the DJI M30T platform over the Amvrakikos Gulf, this framework established a sub-decimeter Ground Sample Distance (GSD) of approximately 4.09 cm/pixel. Evaluating multi-scale spatial configurations demonstrated that while the U-Net 1024×1024 network introduces higher computational demands (772.73 GFlops), it yields optimal mapping

precision, achieving a peak validation F1-score of 0.9589, thus, successfully capturing fine-grained vegetation margins.

To overcome the challenges of monocular camera geolocation estimation, geometric overlaps present in single-frame geolocated marsh vectors, perspective displacement due to object elevation and lens distortion effects, a Probabilistic Multi-View Geolocation framework was implemented using Gaussian-Weighted Rasterization. By accumulating localized spatial probability distributions ($P_{i,j}$) and camera footprint density maps ($D_{i,j}$) into a unified global grid, the system normalizes multi-view confidence scores into a continuous $[0, 1]$ georeferenced matrix ($M_{i,j}$). This method eliminates topological fragmentation, producing, clean, continuous global vectors with sub-meter precision (50×50 cm cell resolution). The resulting high-fidelity spatial maps not only facilitates the detection of fine-grained annual marsh morphodynamic shifts, such as erosion and dilation, but also serve as high-quality ground-truth datasets to train and improve broad-scale satellite remote sensing models for long-term coastal ecosystem conservation.

PMV-Geo is evaluated using a high-level evaluation framework developed to solve the problem where two datasets are temporally and spatially variant, due to natural expansion and erosion. The framework in question, Symmetric Cross-Correlation Evaluation (SCCE), adopts a sliding-window-based technique to create sub-segments of both spatially variant vectors (T_0 and T_1), each sub-segment is paired for both vectors based on how symmetrically similar they are, using the most optimal symmetry scoring technique. Three techniques were evaluated, Zero-Mean Normalized Cross-Correlation (ZNCC), Fast Directional Chamfer Matching (FDCM) and Modified Hausdorff Distance (MHD), where ZNCC performed best in both a synthetic and natural AI-driven dataset. After evaluating and ranking each sub-segment pair, a large concentration of percentiles fell within the 0.7 to 0.95 percentile range, meaning a significant portion of the marsh vectors was structurally stable. Other percentiles with lower symmetry were ignored due to natural marsh growth/erosion relative to the older dataset. Our methodology yielded high geolocation precision with 2.9 to 3.5 meters error. While a 2.9 to 3.5 meter geolocation error in comparison with conventional satellite-imagery methodologies is considered very low, PMV-Geo is developed to produce ultra-high-resolution vectors (theoretical $4\ cm/pxl^2$), thus, geolocation precision down to a centimeter is proprietary. Moreover, DJI M30T supports RTK technology, however, the flight plan to retrieve salt marsh images was conducted without the use of GNSS-RTK technology, resulting to a lower geolocation accuracy. According to an official DJI report, the M30T drone does not natively support high-accuracy mapping, meaning geolocation precision is not guranteed. To overcome this limitation for vegetation mapping, several hardware and methodological alternatives can be adopted. The most direct hardware solution is upgrading to the DJI M350 RTK, which offers extended flight time, superior barometric sensors, and integrated GNSS-RTK technology. If retaining the current hardware, real-time kinematic (RTK) corrections can be achieved either by utilizing a location-based subscription service, connected via a national base station, or by deploying a dedicated DJI D-RTK 3 mobile station on-site for centimeter-level geolocation accuracy. Finally, a soft-ware based alternative, post-processing algorithm can be implemented to retroactively correct the latitudial and longitudial coordinates of the drone, using satellite imagery.

For future work, this application can be further scaled through several methods. Figure 45, Illustrates a design and road-map of several tasks that can be adopted to further scale **PMV-Geo**. This thesis already demonstrates a highly robust methodology to evaluate the geospatial precision of PMV-Geo using the SCCE algorithm. Computer vision is highly versatile enabling

for scalability across multiple domains and fields, but is limited to the precision of Computer Vision models. Thus, for future plans, expanding the coastal marsh dataset into multiple domains and categories can improve precision and model generalization. It currently leverages labels from a single category (Salt Marsh=1, Background=0) and a total of 328 images from a single domain (the Amvrakikos Gulf). Coastal marshes consist of several species with distinct functions, making monitoring and classification critical, thus, Multi-disciplinary assistance and cooperation with professional ecologists, can enhance the dataset into multiple categories on a species-level and domains including Venice Lagoon. Another method to scale PMV-Geo, would be to adopt multi-source monitoring methods (e.g. satellite imagery), to monitor marsh health, carbon sequestration, chl-a levels and tidal events.

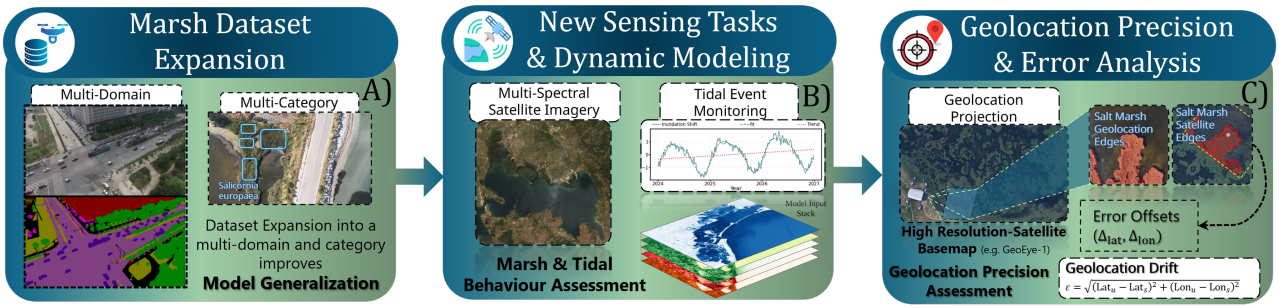


Figure 38: Future Work and Scalability Design of PMV-Geo

5.2 Application 2: Geolocation Estimation of Detected and Tracked Targets in Aerial Footage

This application introduces a robust, precise, real-time framework for the detection, tracking and geolocation estimation of pedestrians and vehicles in low to high altitude Unmanned Aerial Vehicle (UAV) aerial footage. Traditional aerial monitoring workflows rely solely on the task of detection and minimally on the extraction of additional data from the detected objects [93, 94]. A dynamic framework is developed to combine high precision models while integrating the geolocation pipeline for real-time inference on high-resolution aerial footage. Finally, several other additions are made to improve data retrieval from the detections.

5.2.1 Core Contributions

Real-time inference is often challenging in aerial footage where high-resolution imagery, combined with small objects occupying a small portion of the image, is a critical challenge for standard surveillance systems. Moreover, altitude fluctuations aggravates this remote sensing task due to significant object scale variations. In remote sensing tasks, latency and processing overhead are often major limitations for real-time inference. This application proposes a dynamic solution for the multi-scaling problem, applies the geolocation pipeline to extract GPS coordinates in real-time and leverages multi-object tracking models to sustain the GPS coordinates of corresponding detections. Additionally, several visualizations are demonstrated for this application by combining real-time basemap visualizations of the geolocated objects relative to the drone's footprint.

5.2.2 Geolocation Evaluation Framework

To evaluate the geolocation pipeline accuracy, a three-phase plan was developed for the quantitative evaluation of the performance and robustness of the pipeline:

Phase I: Data Acquisition:
This phase focuses on establishing precise ground-truth data to perform ground-truth matching of the system's geolocation estimates. **Ground Control Points** (GCPs) were defined through a **virtual-based GPS marking** method: Physical field testing is limited due to the lack of equipment, thus, an alternative approach is to utilize high-resolution satellite basemaps (e.g., tools Google Earth Pro, QGIS) to define "True" GPS Coordinates, however, this approach is highly prone to human error. To establish True GCPs, videos are collected using the DJI Matrice 30 Thermal drone, to capture vehicles and pedestrians. Based on the positioning of each object, GCPs are created using virtual-based GPS marking tools.

Phase II: Model Robustness and Stress Testing: This phase evaluates the pipeline's resilience under variable operational limits. Controlled flights are designed to systematically isolate and expose specific error sources. For this phase, several scenarios were considered to evaluate how georeferencing accuracy is impacted:

- **Altitude Variations:** Progressively increasing altitudes are leveraged to evaluate the stability of the digital projection matrix. Increasing errors with altitude variations will reveal fundamental issues of the pipeline such as intrinsic parameters, lens distortion and more.
- **Gimbal and UAV Motion:** Rapid angular gimbal variations and UAV movements will enable the monitoring of errors due to extrinsic parameter de-syncing, GPS drifts and more.

Quantitative Metrics:
To precisely quantify the difference between estimated geolocated targets (GPS_{tgt}) and their true coordinates (GPS_{true}), this framework adapts several quantitative metrics to evaluate the performance To quantify the difference between the estimated GPS coordinates and true GPS coordinates, several metrics are leveraged to measure the pipeline's overall error across different scenarios.

- **Haversine Distance [133]:** Computes the distance over the Earth's surface between the true and estimated GPS coordinates. This provides an absolute metric localization error in meters.
- **RMSE (Root Mean Square Error):** Evaluates the overall system performance by providing a measure of the magnitude of the errors. Lower RMSE indicates better performance.

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} d_i^2}$$

Where d_i is the Haversine error of the i-th target detection.

- **CEP (Circular Error Probable):** A statistical metric mapping precision as a radial distance centered around the true GPS coordinate with varying radial distances containing a specified percentage of spatial estimations. The evaluation for this study utilizes CEP@50% (median accuracy) and CEP@95% (maximum operational bounds) to localized target density distributions (Figure 39).

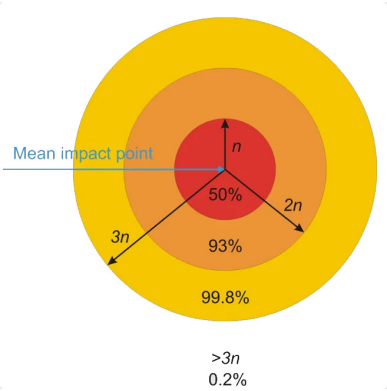


Figure 39: Circular Error Probable (CEP) with Radial Percentages

To determine the Geolocation Estimation framework as a highly robust and optimized pipeline, the CEP@95 metric is ideal for this use-case as it evaluates the furthest points from the ground-truth data point. Thus, achieving low CEP@95 error suggests that the pipeline operates with really high precision even for most outliers.

5.2.3 Data Acquisition & Geolocation Evaluation Scenarios

While public benchmarks such as the VisDrone dataset [78] are effective for training and validating computer vision models, evaluating a coordinate projection framework requires raw sensor metadata and a more alternative approach for evaluation. Consequently, custom aerial videos were captured across the campus of the Department of Informatics and Telecommunications of the University of Ioannina. To systematically evaluate the geolocation pipeline performance, several variables are considered. Two controlled flight scenarios were planned and executed to monitor lens distortion errors, sensor drift and several other inaccuracies. The configuration of each scenario are thoroughly detailed in Table 12.

Scenario	Setup Configuration	Error Sources	Description
Staircase Scenario	**Downward Gimbal, Increasing Elevation, Static Location**	**+Lens Distortion**	**Downward view of objects captured at increasing altitudes from 30m to 120m.**
Non-Static Scenario	**Downward Gimbal, Drone Movement**	**++GPS Drift +GNSS Error +Lens Distortion**	**Continuous drone movement with downward gimbal to evaluate GPS drift and GNSS positioning errors.**

Table 12: Video Scenarios for Robust Geolocation Assessment.

These controlled scenarios enable systematic isolation and quantification of different error sources including lens distortion, camera intrinsic parameter inaccuracies, GPS drift, and GNSS positioning errors. The framework provides a comprehensive foundation for assessing the spatial accuracy and reliability of UAV-based geolocation of detected objects across diverse operational conditions. For the staircase scenario, a single video was captured lasting 56 seconds, with the UAV gimbal pointing downwards ($\theta_{uav} = -90.4°$), while hovering over the university campus. The drone's altitude started from $alt_{uav} = 30m$ and slowly ascended to

$alt_{uav} = 120m$. Throughout the scene, several vehicles were captured with the goal of detecting and subsequently geolocating them. For the Non-static scenario, the UAV was at a lower altitude and scanned an area dense with vehicles around the campus with a similar gimbal position ($\theta_{uav} = -90.4°$) as the staircase scenario.

5.2.4 Training and Detection of Optimized Networks

To detect vehicles and pedestrians for this application, several computer vision models were trained, evaluated and tested under the VisDrone dataset. The best performing models were leveraged for this application based on prior evaluations (Table 5) and were subsequently architecturally improved for edge-based devices and precision. Previously, this thesis expanded on the challenges of Computer Vision in Aerial Imagery (Section 3.1) and a comprehensive breakdown of the VisDrone 2019 dataset [78] (Section 3.3), for this section, we revisit several of these key notes to develop a framework leveraging fine-tuned optimized detection models for real-time and precise inference. As an initial validation and further fine-tuning, YOLO11m, YOLO12m, YOLO26m and YOLO26m-e2e were evaluated.

5.2.5 Detection and Geolocation of Vehicles and Pedestrians

Initially, the YOLO26m-e2e model is employed to automate the detection task for each scenario. Raw video frames are analyzed by the YOLO26m-e2e model, and along with the detector model, a BotSORT tracker [134] is employed to maintain consistent unique identifiers for each vehicle and pedestrian across consecutive frames. To geolocate the detections, each center point of a detection (x_c, y_c) is calculated using the bounding box coordinates produced by the model $(x_{min}, y_{min}, x_{max}, y_{max})$:

$$x_c = \frac{x_{min} + x_{max}}{2} \tag{24}$$

$$y_c = \frac{y_{min} + y_{max}}{2} \tag{25}$$

Once each center point is computed, they are stored in a dictionary with their corresponding label (e.g., pedestrian, vehicle, etc). By leveraging the intrinsic parameters K, each center pixel is translated into physical GPS coordinates relative to GPS_{uav}. Given a detected object's center at $P(u, v)$ (where $u = x_c$ and $v = y_c$), we compute its physical displacement vector relative to the asymmetric principal center (C_x, C_y). By inverting the intrinsic calibration matrix $\mathbf{K}$, we map each pixel center into a normalized Camera Ray Vector using the geolocation pipeline's equation 9. Subsequently, the Camera Ray Vector results are projected into real world gps coordinates by scaling it onto geodetic degrees 15. A visualization of the results produced by the framework was compiled, Figure 40), illustrates several temporal points across the recorded footage, where the drone slowly ascends towards from 30m to 120m. The first row illustrates the footage with detection overlays, along with their corresponding unique identifiers. The second row illustrates a satellite basemap with the drone's localized footprint depicted as the yellow dotted lines, the blue 'X' marker, the drone, and the red labels, the geolocated detections with their corresponding label and identifier.

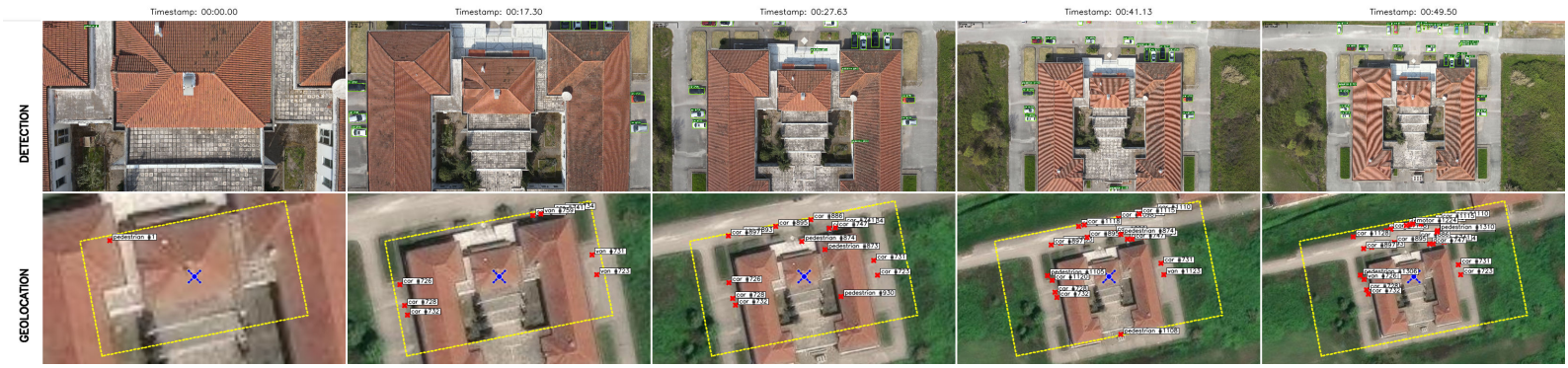


Figure 40: Detection and Geolocation Composite illustrating BBox Predictions, Drone Footprint and the projected Predictions with their corresponding labels

In computer vision and remote sensing pipelines, a behaviour during inference is observed where objects going in and out of frame results to occlusion bounding box drifting. This unique problem where the detector's coordinates actively drifts or expands until the entire object is visible. In academic literature regarding aerial imagery, this is formally known as Centroid Drift induced by partial Boundary Truncation [135]. As a result, because the Areal Drifting Centroid shifts in pixel coordinates, this directly impacts the projection pipeline, resulting in GPS drift. This example is perfectly illustrated in the acquired data, Figure 41, with shifting visible centroids.



Figure 41: Centroid Drift induced by Partial Boundary Truncation

To isolate projection errors introduced by Partial Boundary Truncation during target entry and exit transitions, a dynamic geometric filter was applied. One study proposed the Within-Bounds (WB) metric [136] to estimate as an alternative quantitative metric and identify false/true positives and negatives. The WB metric was adopted to the evaluation protocols of the geolocation estimation framework to demonstrate the Centroid Shifting effects during target entry and exit.

5.2.6 Geolocation Evaluation Results

To evaluate the precision of the proposed projection pipeline under varying camera altitudes and drone movements, a systematic metrological analysis on the drone trajectory was adopted. Targets of interest (e.g., pedestrians, vehicles) were continuously tracked, and their geolocated

coordinates were projected into the WGS84 coordinate system using the updated intrinsic parameters. The projected positions were further validated against true ground-truth coordinates obtained through virtual-based GPS labeling tools. For the first evaluation, the UAV executed a continuous vertical ascent from an initial altitude of 30 m to a maximum altitude of 120 m above ground level, a total of 21 ground-truth labels were defined for this scenario. Two evaluation protocols are directly compared, to demonstrate the significance of the Boundary Truncation effect during target entry and exit:

1. **Not Within-Bounds (Not-WB):** Evaluates all frame-level target matches regardless of their physical location relative to the camera viewport boundary.

2. **Within-Bounds (WB):** Restricts evaluation per each target strictly to frames where the ground-truth target coordinates fall inside the camera's dynamic footprint.

Subsequently, to observe projection errors resulted by the reported intrinsic parameters of the drone, such as focal length, mathematical errors, the image was evaluated across three radial subsets starting from the center point of the image. Each subset covered different radial percentages relative to the center point of the image ($\frac{Width}{2}$ and $\frac{Height}{2}$): Inner radial bin, thresholding approximately 0.00% to 0.35, a middle radial bin, $0.35to0.70$ and an outer radial bin $0.70, 1.00$.

Table 13 summarizes the stratified geolocation accuracy under both protocols, reporting the Root Mean Square Error (RMSE), the 50th percentile Circular Error Probable (CEP50), and the 95th percentile Circular Error Probable (CEP95).

Table 13: Altitude Test Geolocation Error Within-Bounds (WB) vs. Non-Within-Bounds (Non-WB)

	RMSE (m)		**CEP50 (m)**		**CEP95 (m)**	
Altitude Interval	**WB**	**Not-WB**	**WB**	**Not-WB**	**WB**	**Not-WB**
30 m – 55 m	1.358	1.844	1.239	1.611	1.713	2.963
55 m – 80 m	1.091	1.156	1.061	1.091	1.472	1.620
80 m – 105 m	0.659	0.776	0.564	0.597	1.104	1.428
105 m – 120 m	0.268	0.268	0.241	0.241	0.449	0.449
Overall Horizon	**0.787**	**0.951**	**0.640**	**0.721**	**1.389**	**1.762**

The comparative results in Table 13 highlight the critical impact of target clipping at lower flight altitudes. In the 30 m – 55 m interval, the Non-WB protocol yields an RMSE of 1.844 m and a CEP95 of 2.963 m, which is significantly higher than the WB RMSE of 1.358 m and CEP95 of 1.713 m. When a target is partially clipped from the image boundaries during entry or exit, its detected bounding box centroid deviates from its true geographical center, inducing additional projection error. By adopting the WB metric over the geolocated objects, centroid drift is significantly mitigated. Additionally, another behaviour observed throughout the evaluation, projection error gradually drops in relation to the ascension of the UAV, converging at high altitudes (105 m – 120 m). This suggests that with each target slowly approaching the camera lens principal point, lens distortion effects are diminishing. Figure 42, illustrates the target centroid trajectories slowly moving towards the image center with increasing altitudes.

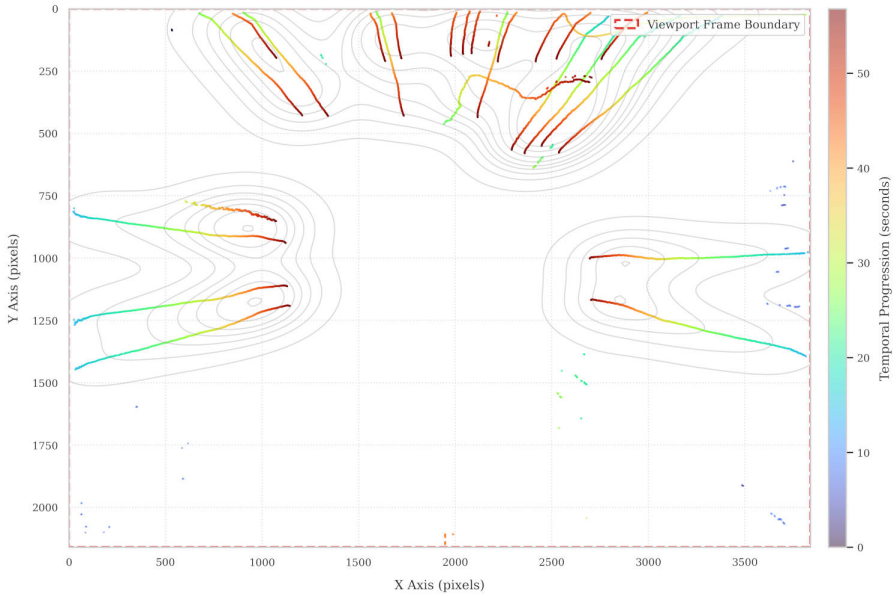


Figure 42: Target Centroid Trajectory & Density Heatmap in Relation to Camera Lens Principal Point

For the second evaluation under the non-static profile, the UAV executed a continuous motion flight path, hovering over a roadway populated with active vehicles and motorcycles. This scenario was conducted at an approximate relative altitude of 45 meters, with a total of 25 ground-truth labels assigned to establish baseline accuracy. During the initial assessment, a persistent issue was observed: the entire image projection was systematically shifted to the right. When evaluated using the original principal point ($PP_CX = 1900$), this misalignment resulted in significant radial and overall positioning errors. This behavior strongly indicated the presence of a localized, systematic Global Navigation Satellite System (GNSS) error. To correct this significant projection error, the principal point was manually shifted 200 pixels to the right ($PP_CX = 2100$). This parameter adjustment shifted the projection towards the opposite direction, thus, simulating a GPS shift, yielding drastic improvements in projection precision. Figure 43 illustrates this projection shift

(a) GNSS Error

(b) Principal Point Shift by 200 pixels



Figure 43: Image Projection Illustrating the GNSS Error

Based on the UAV's Ground Sample Distance (GSD) of approximately 1.67 cm/pixel, the shift was quantified to approximately 3.34 meters on the ground. The Tables 14, 15, quantifies the direct impact of offsetting the principal point on the Within-Bounds (WB) geolocation metrics.

Table 14: Original Principal Point ($PP_CX = 1900$) Geolocation Error Within-Bounds (WB) vs. Not-Within-Bounds (Not-WB)

	RMSE (m)		**CEP50 (m)**		**CEP95 (m)**	
Radial Bin	**WB**	**Not-WB**	**WB**	**Not-WB**	**WB**	**Not-WB**
Inner [0.00, 0.35]	2.646	2.646	2.732	2.732	2.944	2.944
Middle (0.35, 0.70]	2.566	2.560	2.551	2.550	2.970	2.968
Outer (0.70, 1.00]	2.029	2.058	1.916	1.921	2.836	2.873
Overall	**2.469**	**2.468**	**2.558**	**2.558**	**2.956**	**2.956**

Table 15: Shifted Principal Point ($PP_CX = 2100$) Geolocation Error Within-Bounds (WB) vs. Not-Within-Bounds (Not-WB)

	RMSE (m)		**CEP50 (m)**		**CEP95 (m)**	
Radial Bin	**WB**	**Not-WB**	**WB**	**Not-WB**	**WB**	**Not-WB**
Inner [0.00, 0.35]	0.954	0.954	0.771	0.771	1.764	1.764
Middle (0.35, 0.70]	1.256	1.261	1.132	1.135	2.342	2.354
Outer (0.70, 1.00]	1.779	1.812	1.602	1.609	2.771	2.822
Overall	**1.155**	**1.165**	**0.924**	**0.929**	**2.297**	**2.339**

In the temporal evaluation, the differences between the WB and Non-WB filters are marginal. By accounting for the 3.34 meter GPS error, the overall RMSE improved by over 113%, dropping from 2.469m to 1.155m for WB targets. Moreover, a notable trend within the shifted PP

metrics is the errors for each radial bin distribution. While the inner and middle radial bins exhibited high projection precision WB RMSE of 0.954m and 1.256m, respectively, the outer radial bin exhibits significantly higher WB RMSE of 1.1779m. This is projection degradation across the outer radial bins is likely attributed to factors such as radial lens distortion. Unfortunately, DJI, does not provide the radial and tangent distortion coefficients for the 16:9 image, thus, radial distortion will be an immediate source of error for geolocation. To further refine projection accuracy after addressing the horizontal GSS drift for the given scenario, a vertical correction by adjusting the relative altitude by an offset revealed greater performance for projection accuracy. This typically stems from barometric altimetter drift [137] and takeoff elevation missmatch. By lowering the assumed ground plane by 2.5 meters, the shape of the projection becomes geometrically smaller. While the previous results are performed across the entire scenario, where GPS estimates drift, comparing the track-level median estimates reveals a significantly more precise results. The tracker's unique identifiers for each tracked object, can be utilized to extract a singular, refined, GPS coordinate for each target. More specifically, the median GPS estimate across a object's tracking lifespan. This method immediately eliminates the need to use the WB metric. As shown in the comprehensive summary table below 16, stacking each of these empirically determined offsets, along with the projection yielded significantly higher precision across the board.

Table 16: Overall Geolocation Error Comparison: Original vs. Shifted vs. Shifted with Altitude Offset

	RMSE (m)		**CEP50 (m)**		**CEP95 (m)**	
Configuration	**WB**	**Not-WB**	**WB**	**Not-WB**	**WB**	**Not-WB**
Original ($PP_CX = 1900$)	2.429	2.429	2.573	2.573	2.977	2.977
Shifted ($PP_CX = 2100$)	0.933	0.916	0.498	0.496	1.903	1.708
Shifted + Alt Offset ($-2.5\,\mathrm{m}$)	0.498	0.510	0.395	0.422	0.698	0.777

Through these empirically determined offsets and the track-level estimations, the geolocation framework yielded dramatically higher results, achieving an all time low track-level WB RMSE of 0.498, WB CEP@50 0.395 and an incredibly low WB CEP@95 0.698 error, respectively. Notably, these empirically determined offsets were specific to the non-static scenario, and did not yield better results for other scenarios, thus, highlighting the impact of sensor error. Figure 44 illustrates the track-level median GPS estimates with incredibly high precision.



Figure 44: Track Level Median Geolocation Estimates

5.2.7 Application II: Conclusion

While geolocation estimation precision was improved, achieving approximately 0.5m error, several offsets were applied to correct and improve precision across the board. Initially, a correction to the principal point was applied (200 pixels or approximately 3.34 m), simulating a GPS coordinate shift towards the right to align the image projection with the google map background. This drastically reduced geospatial error from ≈ 2.5 m error to $\approx 0.5 - 1$ m error. Moreover, a altimeter barometric correction of -2.5 m reduced geolocation error down to $0.4 - 0.5$ m error. These corrections highlight the importance of adopting Real-time Kinematics (RTK) technology during precise mapping of objects and locations. The current application leveraging Computer Vision to track objects adopts a Within-Bounds (WB) metric, that effectively isolates tracks that have appeared in and out of frame. This metric mitigated GPS drift resulted by object occlusion when bounding box centroids shifted until the full object was visible within the frame (Figure 41). Using WB, geospatial precision improved slightly, particularly on tracked objects across the outer boundaries of the frame.

5.3 Concludion & Discussion

UAV-based geolocation estimation along with computer vision is a workflow that can be scaled with multiple applications, including vegetation mapping and target monitoring. The versatility of Computer Vision, enables these applications to be scaled across several domains through geolocation. This thesis explored these concepts theoretically, systematically and through practical implementations while adopting robust evaluation frameworks. Each application integrates deep learning for computer vision tasks and geolocation estimation to detect boundaries of objects and project them into coordinate systems, with a thorough attention to evaluating the geospatial precision of Geolocation. The evaluation findings of geolocation estimation were quite compelling, with both application exhibiting decent geospatial precision, however, a large geospatial error originated from the lack of Real-Time Kinematic (RTK) during the conducted flight plans. Both applications without additional fine-tuning achieved a displacement error

of 2.5 to 3.5 m meters that stems from using GNSS. However, for precise geolocation tasks, 2.5 - 3.5 m error is really high and can be eliminated by enabling RTK, or improved through Post-Processing Kinematic methods.

The first application focused on the development of an end-to-end framework titled **Probabilistic Multi-View Geolocation** (PMV-Geo) that processes multiple UAV Aerial images of salt marshes to semantically segment them using *YOLO26x-sem* project multiple semantic masks into a unified ultra-high-resolution marsh vector. For this application, a custom computer vision salt marsh dataset was developed of the Amvrakikos Gulf, YOLO26x was evaluated and compared with a conventional U-Net architecture and achieved better performance across all metrics than U-Net. Moreover, for the Geospatial evaluation of PMV-Geo, two types of vectors were utilized a T_0 and T_1. The T_0 vector, corresponds to the high-resolution satellite image (GeoEye-1 Google satellite) labeled using Sam 2.1 for guided semantic segmentation with negative/positive point prompts, thus, producing a 1:1 geospatial marsh vector for the date April 4, 2024. The T_1 vector, corresponds to the prediction results of PMV-Geo, a Unified Geospatial vector of salt marshes, produced through multiple UAV aerial imagery retrieved for the date July 17, 2025. Due to the high temporal variance of both vectors being 467 days apart, including the extremely dynamic behaviour of salt marshes expanding and eroded within short time-periods, a specialized evaluation framework was developed titled Symmetric Cross-Correlation Evaluation (SCCE). The framework effectively isolated false positive geometries resulted by newly eroded or expanded marsh bodies, thus, prioritizing in core marsh bodies that have remained physically the same throughout both vectors. SCCE, exhibited a total geospatial accuracy of 2.90 - 3.17 m of Mean Displacement. The second application focused primarily on the detection of bounding box boundaries of objects such as vehicles through UAV aerial footage. This application integrates the YOLO26x object detector [75], optimized for small target detection and a the BotSort tracker [134] to sustain the same numerical identifiers for each object across multiple frames. The application was adopted with cropped 16:9 images, that impacted geolocation accuracy due to intrinsic parameter specificity and GNSS error. After offsetting several geolocation parameters with the goal of minimizing GNSS error, identifying the true Principal Point of the cropped 16:9 image, and reducing barometric altimeter error, the total geospatial precision achieved throughout the application was as low as 0.395 meters using the Within-Bound metric for the CEP50 metric, whereas the CEP95 metric achieved an all time low of 0.698 meters, which validates the high precision even for outlier boundaries. Overall, both applications exhibit high precision for geolocation estimation, but are limited to several factors including lack of depth and enabled RTK technology.

Several future plans to continue adopting new techniques in this thesis includes solving the lack of RTK precision and adopt Monocular Depth Estimation using conventional Gaussian Splatting and 3D Reconstruction methodologies. If the integration of 3D Reconstruction is adopted into the framework, along with Computer Vision and Geolocation Estimation, it will be the first time to ever implement a UAV Geolocation mapping application with full monocular depth, thus, minimizing geospatial error even from object elevation. Furthermore, the first application does not require 3D reconstruction, due to the distinct object elevation of coastal marshes, however, several future plans for PMV-Geo are in progress.

The PMV-Geo application can be further scaled through several methods. Figure 45, Illustrates a design and road-map of several tasks that can be adopted to further scale it. This thesis already demonstrates a highly robust methodology to evaluate the geospatial precision of PMV-Geo using the SCCE algorithm. Computer vision is highly versatile enabling for scal-

ability across multiple domains and fields, but is limited to the precision of Computer Vision models. Thus, for future plans, expanding the coastal marsh dataset into multiple domains and categories can improve precision and model generalization. It currently leverages labels from a single category (Salt Marsh=1, Background=0) and a total of 328 images from a single domain (the Amvrakikos Gulf). Coastal marshes consist of several species with distinct functions, making monitoring and classification critical, thus, Multi-disciplinary assistance and cooperation with professional ecologists, can enhance the dataset into multiple categories on a species-level and domains including Venice Lagoon. Another method to scale PMV-Geo, would be to adopt multi-source monitoring methods (e.g. satellite imagery), to monitor marsh health, carbon sequestration, chl-a levels and tidal events.

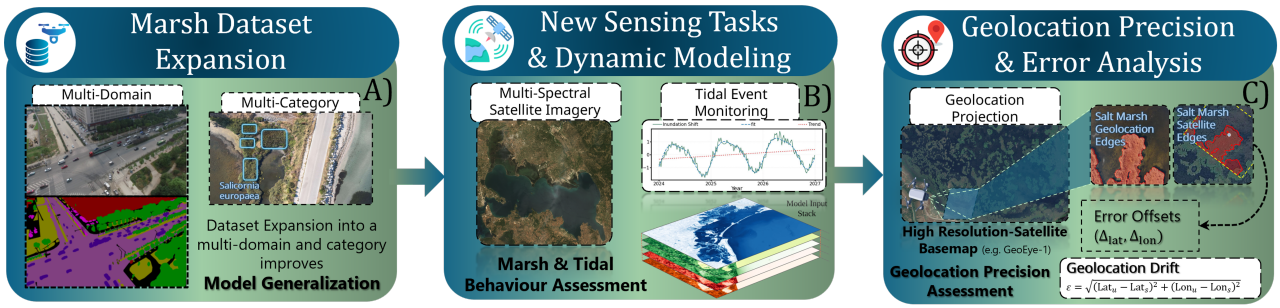


Figure 45: Future Work and Scalability Design of PMV-Geo

Future plans regarding the second application, includes adopting 3D Reconstruction methodologies through monocular cameras. The goal is to capture images of objects from multiple perspectives and adopt Gaussian splatting. Through post-processing geolocation can be optimized to include the elevation of objects. Moreover, the framework can be further enhanced to construct 3D environments of the scanned areas.

References

[1] R. EDWARDS, "Chapter 22 - the eye. defects of vision and optical instruments," in *Physics for O.N.C. Courses*, pp. 392–419, Pergamon, 1970.

[2] S. Verykokou and C. Ioannidis, "Oblique aerial images: Geometric principles, relationships and definitions," *Encyclopedia*, vol. 4, no. 1, pp. 234–255, 2024.

[3] P. Vangorp, C. Richardt, E. A. Cooper, G. Chaurasia, M. S. Banks, and G. Drettakis, "Perception of perspective distortions in image-based rendering," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 4, pp. 1–12, 2013.

[4] B. Kerbl, G. Kopanas, T. Leimkühler, G. Drettakis, *et al.*, "3d gaussian splatting for real-time radiance field rendering.," *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[5] S.-H. Lee, J.-Y. Lee, and J.-S. Choi, "Lens distortion correction using a checkerboard pattern," in *Proceedings of the 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pp. 1–2, 2008.

[6] R. Van Alphen, K. C. Rains, M. Rodgers, R. Malservisi, and T. H. Dixon, "Uav-based wetland monitoring: Multispectral and lidar fusion with random forest classification," *Drones*, vol. 8, no. 3, p. 113, 2024.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR, abs/1506.02640*, 2015.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," 2015.

[9] D. Mpouziotas, E. Mastrapas, N. Dimokas, P. Karvelis, and E. Glavas, "Object detection for low light images," in *2022 7th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, pp. 1–6, 2022.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[11] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 06, no. 02, pp. 107–116, 1998.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition. corr abs/1512.03385 (2015)," 2015.

[13] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," 2018.

[14] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," 2021.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, pp. 213–229, Springer, 2020.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[17] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *International Journal of Computer Vision*, vol. 128, p. 336–359, Oct. 2019.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[20] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," 2016.

[21] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[22] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[23] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[24] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," 2019.

[25] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," 2020.

[26] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2018.

[27] H. Gholamalinezhad and H. Khosravi, "Pooling methods in deep neural networks, a review," *arXiv preprint arXiv:2009.07485*, 2020.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[29] A. F. Agarap, "Deep learning using rectified linear units (relu)," 2019.

[30] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973. The Sobel operator is credited to I. Sobel.

[31] A. L. Maas, A. Y. Hannun, A. Y. Ng, *et al.*, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, vol. 30, p. 3, Atlanta, GA, 2013.

[32] K. W. Goh, S. Surono, M. F. Afiatin, K. R. Mahmudah, N. Irsalinda, M. Chaimanee, and C. W. Onn, "Comparison of activation functions in convolutional neural network for poisson noisy image classification," *Emerg Sci J*, vol. 8, no. 2, pp. 592–602, 2024.

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[36] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

[38] J. Devlin, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[39] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

[40] A. Kaya, I. Bilik, and I. Stainvas, "A comparative study of vision transformers and cnns for few-shot rigid transformation and fundamental matrix estimation," *arXiv preprint arXiv:2510.04794*, 2025.

[41] J. Maurício, I. Domingues, and J. Bernardino, "Comparing vision transformers and convolutional neural networks for image classification: A literature review," *Applied Sciences*, vol. 13, no. 9, p. 5521, 2023.

[42] S. Takahashi, Y. Sakaguchi, N. Kouno, K. Takasawa, K. Ishizu, Y. Akagi, R. Aoyama, N. Teraya, A. Bolatkan, N. Shinkai, *et al.*, "Comparison of vision transformers and convolutional neural networks in medical image analysis: a systematic review," *Journal of Medical Systems*, vol. 48, no. 1, p. 84, 2024.

[43] Y. Li, S. Xie, X. Chen, P. Dollar, K. He, and R. Girshick, "Benchmarking detection transfer learning with vision transformers," *arXiv preprint arXiv:2111.11429*, 2021.

[44] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[45] C. Wang, H. Liao, I. Yeh, Y. Wu, P. Chen, and J. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn corr," *vol. abs/1911.11929*, 2019.

[46] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7464–7475, 2023.

[47] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," *arXiv preprint arXiv:2211.04800*, 2022.

[48] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023. Software, License: AGPL-3.0.

[49] Y. Tian, Q. Ye, and D. Doermann, "Yolov12: Attention-centric real-time object detectors," 2025.

[50] Z. Zong, G. Song, and Y. Liu, "Detrs with collaborative hybrid assignments training," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6748–6758, 2023.

[51] Y. Peng and Y. Wang, "Cnn and transformer framework for insect pest classification," *Ecological Informatics*, vol. 72, p. 101846, 2022.

[52] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "Vqa: Visual question answering," in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

[53] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

[54] J. H. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6469–6477, 2017.

[55] X. Du, T.-Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q. V. Le, and X. Song, "Spinenet: Learning scale-permuted backbone for recognition and localization," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11592–11601, 2020.

[56] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "Cspnet: A new backbone that can enhance learning capability of cnn," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 390–391, 2020.

[57] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, 2016.

[58] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," in *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 13029–13038, 2021.

[59] J. Redmon, "Darknet: Open source neural networks in c." `https://pjreddie.com/darknet/`, 2013–2016.

[60] S. Carata, R. Mihaescu, E. Barnoviciu, M. Chindea, M. Ghenescu, and V. Ghenescu, "Complete visualisation, network modeling and training, web based tool, for the yolo deep neural network model in the darknet framework," in *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 517–523, 2019.

[61] J. Ansel, E. Yang, and et al, "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation," in *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*, ACM, Apr. 2024.

[62] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," *arXiv preprint arXiv:2211.04800*, 2022.

[63] G. Jocher, A. Chaurasia, and et al, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," 22nd of November.

[64] P. Jaccard, "Etude comparative de la distribution florale dans une portion des alpes et des jura,"

[65] R. Khanam and M. Hussain, "What is yolov5: A deep look into the internal features of the popular object detector," *arXiv preprint arXiv:2407.20892*, 2024.

[66] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," *arXiv preprint arXiv:1703.00810*, 2017.

[67] J. Li, P. Lefevre, and A. P. A. Majeed, "Randomness and interpolation improve gradient descent: A simple exploration in cifar datasets," in *2024 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 56–59, IEEE, 2024.

[68] A. M. Saxe, Y. Bansal, J. Dapello, M. Advani, A. Kolchinsky, B. D. Tracey, and D. D. Cox, "On the information bottleneck theory of deep learning," in *International Conference on Learning Representations*, 2018.

[69] C.-Y. Wang, I.-H. Yeh, and H.-Y. Mark Liao, "Yolov9: Learning what you want to learn using programmable gradient information," in *European conference on computer vision*, pp. 1–21, Springer, 2024.

[70] X. Zhang, H. Zeng, S. Guo, and L. Zhang, "Efficient long-range attention network for image super-resolution," in *European conference on computer vision*, pp. 649–667, Springer, 2022.

[71] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," *Advances in neural information processing systems*, vol. 37, pp. 107984–108011, 2024.

[72] R. Sapkota, Z. Meng, M. Churuvija, X. Du, Z. Ma, and M. Karkee, "Comprehensive performance evaluation of yolov12, yolo11, yolov10, yolov9 and yolov8 on detecting and counting fruitlet in complex orchard environments," *Agriculture Communications*, p. 100125, 2026.

[73] R. Khanam and M. Hussain, "Yolov11: An overview of the key architectural enhancements," *arXiv preprint arXiv:2410.17725*, 2024.

[74] M. Huang and W. Jiang, "Dms-yolo: Small target detection algorithm based on yolov11," *PLoS One*, vol. 21, no. 1, p. e0341991, 2026.

[75] R. Sapkota, R. H. Cheppally, A. Sharda, and M. Karkee, "Yolo26: key architectural enhancements and performance benchmarking for real-time object detection," *arXiv preprint arXiv:2509.25164*, 2025.

[76] M. Lei, S. Li, Y. Wu, H. Hu, Y. Zhou, X. Zheng, G. Ding, S. Du, Z. Wu, and Y. Gao, "Yolov13: Real-time object detection with hypergraph-enhanced adaptive visual perception," *arXiv preprint arXiv:2506.17733*, 2025.

[77] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[78] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

[79] J. Liu, J. Su, X. Yao, Z. Jiang, G. Lai, Y. Du, Y. Qin, W. Xu, E. Lu, J. Yan, *et al.*, "Muon is scalable for llm training," *arXiv preprint arXiv:2502.16982*, 2025.

[80] G. Schulz, "Iterative berechung der reziproken matrix," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 13, no. 1, pp. 57–59, 1933.

[81] A. S. Bhadoriya, V. Vegamoor, and S. Rathinam, "Vehicle detection and tracking using thermal cameras in adverse visibility conditions," *Sensors*, vol. 22, no. 12, p. 4567, 2022.

[82] A. S. Chakravarthy, S. Sinha, P. Narang, M. Mandal, V. Chamola, and F. Yu, "Dronesegnet: Robust aerial semantic segmentation for uav-based iot applications," *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 4277–4286, 2022.

[83] A. Calantropio, F. Chiabrando, M. Codastefano, and E. Bourke, "Deep learning for automatic building damage assessment: Application in post-disaster scenarios using uav data," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. V-1-2021, pp. 113–120, 2021.

[84] D. Mpouziotas, P. Karvelis, P. Georgiou, G. Pachoulas, and C. Stylios, "End-to-end web-gis and computer vision system using uav imagery for avian monitoring and risk assessment," in *2025 10th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM)*, (Arta, Greece), IEEE, 2025.

[85] S. Tan, Z. Duan, and L. Pu, "Multi-scale object detection in uav images based on adaptive feature fusion," *Plos one*, vol. 19, no. 3, p. e0300120, 2024.

[86] X. Wang, Y. Xu, X. Fu, and Z.-J. Zha, "Mgaw: An effective method for geo-localization in adverse weather," in *Proceedings of the 2nd Workshop on UAVs in Multimedia: Capturing the World from a New Perspective*, pp. 19–23, 2024.

[87] B. Liu, J. Jin, Y. Zhang, and C. Sun, "Wrrt-detr: weather-robust rt-detr for drone-view object detection in adverse weather," *Drones*, vol. 9, no. 5, p. 369, 2025.

[88] T. Yue, X. Lu, J. Cai, Y. Chen, and S. Chu, "Yolo-mst: Multiscale deep learning method for infrared small target detection based on super-resolution and yolo," *Optics & Laser Technology*, vol. 187, p. 112835, 2025.

[89] D. Mpouziotas, P. Karvelis, and C. Stylios, "Advanced computer vision methods for tracking wild birds from drone footage," *Drones*, vol. 8, no. 6, p. 259, 2024.

[90] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing aided hyper inference and fine-tuning for small object detection," in *2022 IEEE international conference on image processing (ICIP)*, pp. 966–970, IEEE, 2022.

[91] D. Mpouziotas, P. Karvelis, I. Tsoulos, and C. Stylios, "Automated wildlife bird detection from drone footage using computer vision techniques," *Applied Sciences*, vol. 13, no. 13, p. 7787, 2023.

[92] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

[93] C. Wei and W. Wang, "Rfag-yolo: A receptive field attention-guided yolo network for small-object detection in uav images," *Sensors*, vol. 25, no. 7, p. 2193, 2025.

[94] X. Liu, Y. Zheng, Y. Cai, Y. Ding, J. Li, W. Kang, and Z. Cai, "Rfha-yolo: Dynamic receptive field and adaptive hybrid attention for small-object detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2026.

[95] Y. Lyu, G. Vosselman, G.-S. Xia, A. Yilmaz, and M. Y. Yang, "Uavid: A semantic segmentation dataset for uav imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 165, pp. 108 – 119, 2020.

[96] M. Rahnemoonfar, T. Chowdhury, A. Sarkar, D. Varshney, M. Yari, and R. Murphy, "Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. arxiv," *arXiv preprint arXiv:2012.02951*, 2020.

[97] D. Du, Y. Qi, H. Yu, Y. Yang, K. Duan, G. Li, W. Zhang, Q. Huang, and Q. Tian, "The unmanned aerial vehicle benchmark: Object detection and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 370–386, 2018.

[98] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[99] T. Yue, X. Lu, J. Cai, Y. Chen, and S. Chu, "Yolo-mst: Multiscale deep learning method for infrared small target detection based on super-resolution and yolo," *Optics & Laser Technology*, vol. 187, p. 112835, 2025.

[100] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7373–7382, 2021.

[101] Y. Lu, Y. Lin, H. Wu, X. Xian, Y. Shi, and L. Lin, "Sirst-5k: Exploring massive negatives synthesis with self-supervised learning for robust infrared small target detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 62, pp. 1–11, 2024.

[102] U. S. D. M. Agency, *Department of Defense World Geodetic System 1984: its definition and relationships with local geodetic systems*, vol. 8350. Defense Mapping Agency, 1987.

[103] L. Yinhong, M. Zhijun, G. Wang, L. Mingkai, and Y. Enqi, "Review of research on satellite clock bias prediction models in gnss," *Remote Sensing*, vol. 17, no. 18, p. 3177, 2025.

[104] L. Heng, G. X. Gao, T. Walter, and P. Enge, "Statistical characterization of glonass broadcast ephemeris errors," in *Proceedings of the 24th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS 2011)*, (Portland, OR), pp. 3109–3117, Institute of Navigation, Sept. 2011.

[105] M. M. Douglas, L. Campbell, and V. Klemas, "The ecological importance of salt marshes: Carbon sequestration, biodiversity and coastal protection," *Estuarine, Coastal and Shelf Science*, vol. 178, pp. 35–42, 2016.

[106] E. Mcleod, G. L. Chmura, S. Bouillon, R. Salm, M. Björk, C. M. Duarte, C. E. Lovelock, W. H. Schlesinger, and B. R. Silliman, "A blueprint for blue carbon: toward an improved understanding of the role of vegetated coastal habitats in sequestering co2," *Frontiers in Ecology and the Environment*, vol. 9, no. 10, pp. 552–560, 2011.

[107] X. Ouyang and S. Y. Lee, "Updated estimates of carbon accumulation rates in coastal marsh sediments," *Biogeosciences*, vol. 11, no. 18, pp. 5057–5071, 2014.

[108] P. S. E. zu Ermgassen, J. H. Grabowski, J. R. Gair, and S. P. Powers, "Quantifying fish and mobile invertebrate production from a threatened nursery habitat," *Journal of Applied Ecology*, vol. 53, no. 2, pp. 596–606, 2016.

[109] J. Lee, P. zu Ermgassen, B. DeAngelis, M.-S. Lee, A. R. Hughes, and J. H. Grabowski, "Habitat characteristics drive fish recruitment enhancement in threatened coastal nursery habitats," *Journal of Applied Ecology*, vol. 63, no. 5, p. e70360, 2026. e70360 JAPPL-2025-01760.R1.

[110] A. Himes-Cornell, L. Pendleton, and P. Atiyah, "Valuing ecosystem services from blue forests: A systematic review of the valuation of salt marshes, sea grass beds and mangrove forests," *Ecosystem services*, vol. 30, pp. 36–48, 2018.

[111] K. Elschot, M. J. Baptist, and M. E. B. van Puijenbroek, "Biocompacting livestock accelerate drowning of tidal salt marshes with sea level rise," *Frontiers in Marine Science*, vol. Volume 10 - 2023, 2023.

[112] E. B. Watson, C. Wigand, E. W. Davey, H. M. Andrews, J. Bishop, and K. B. Raposa, "Wetland loss patterns and inundation-productivity relationships prognosticate widespread salt marsh loss for southern new england," *Estuaries and Coasts*, vol. 40, pp. 662–681, May 2017.

[113] NOAA Office for Coastal Management, *Standard Operating Procedures: Mapping Land Use and Habitat Change in the National Estuarine Research Reserve System*. National Oceanic and Atmospheric Administration, 2022.

[114] C. Zhang, D. R. Mishra, and S. C. Pennings, "Mapping salt marsh soil properties using imaging spectroscopy," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 148, pp. 221–234, 2019.

[115] R. Xu, Y. Fan, B. Fan, G. Feng, and R. Li, "Classification and monitoring of salt marsh vegetation in the yellow river delta based on multi-source remote sensing data fusion," *Sensors*, vol. 25, no. 2, 2025.

[116] J. Zhen, D. Mao, Z. Shen, D. Zhao, Y. Xu, J. Wang, M. Jia, Z. Wang, and C. Ren, "Performance of xgboost ensemble learning algorithm for mangrove species classification with multisource spaceborne remote sensing data," *Journal of Remote Sensing*, vol. 4, p. 0146, 2024.

[117] A. Collin, N. Lambert, and S. Etienne, "Satellite-based salt marsh elevation, vegetation height, and species composition mapping using the superspectral worldview-3 imagery," *International Journal of Remote Sensing*, vol. 39, no. 17, pp. 5619–5637, 2018.

[118] M. Schmitt, L. H. Hughes, C. Qiu, and X. X. Zhu, "Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion," *arXiv preprint arXiv:1906.07789*, 2019.

[119] A. Sekertekin, A. Marangoz, and H. Akcin, "Pixel-based classification analysis of land use land cover using sentinel-2 and landsat-8 data," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 42, pp. 91–93, 2017.

[120] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[121] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, *et al.*, "Sam 2: Segment anything in images and videos, 2024," *URL https://arxiv. org/abs/2408.00714*, vol. 3, 2024.

[122] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[123] A. Mao, M. Mohri, and Y. Zhong, "Cross-entropy loss functions: Theoretical analysis and applications," in *International conference on Machine learning*, pp. 23803–23828, pmlr, 2023.

[124] J. N. K. Poo, M. S. Saealal, M. Z. Ibrahim, and M. Yakno, "Leveraging u-net architecture for accurate localization in brain tumor segmentation," in *2023 IEEE 9th Information Technology International Seminar (ITIS)*, pp. 1–6, IEEE, 2023.

[125] P. Ranganathan and E. Olson, "Gaussian process for lens distortion modeling," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3620–3625, 2012.

[126] Esri, "World Imagery Basemap," 2026. (Accessed on 2026-01-28).

[127] Z. Fu, W. Song, Y. Wang, X. Wu, Y. Zheng, Y. Zhang, D. Xu, X. Wei, T. Xu, and X. Zhao, "Sliding window attention training for efficient large language models," *arXiv preprint arXiv:2502.18845*, 2025.

[128] G. A. Ayubi, B. Kowalski, and A. Dubra, "Normalized weighted cross correlation for multi-channel image registration," *Optics continuum*, vol. 3, no. 5, pp. 649–665, 2024.

[129] S. Lee, D. Kim, I. Park, G. Kim, and S. Kim, "Perceptible lightweight zero-mean normalized cross-correlation for infrared template matching," *IEEE Access*, vol. 12, pp. 164777–164791, 2024.

[130] *Standard Deviation*, pp. 505–508. New York, NY: Springer New York, 2008.

[131] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, and R. Chellappa, "Fast directional chamfer matching," in *2010 IEEE Computer Society conference on computer vision and pattern recognition*, pp. 1696–1703, IEEE, 2010.

[132] M.-P. Dubuisson and A. Jain, "A modified hausdorff distance for object matching," in *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, pp. 566–568 vol.1, 1994.

[133] R. Agramanisti Azdy and F. Darnis, "Use of haversine formula in finding distance between temporary shelter and waste end processing sites," in *Journal of Physics: Conference Series*, vol. 1500, p. 012104, IOP Publishing, 2020.

[134] N. Aharon, R. Orfaig, and B.-Z. Bobrovsky, "Bot-sort: Robust associations multi-pedestrian tracking," *arXiv preprint arXiv:2206.14651*, 2022.

[135] D. Miao, X. He, B. Zhang, P. Li, X. Cheng, M. Qiao, H. Zhao, Z. Zheng, and G. Yang, "A unified framework for scale-adaptive and truncation-aware remote sensing object detection," *Measurement Science and Technology*, 2026.

[136] R. Sawahn and M. Schall, "Falconeye: Efficient centroid-based object detection for maritime high altitude uav images on embedded devices," in *Proceedings of the Winter Conference on Applications of Computer Vision*, pp. 1617–1628, 2025.

[137] T. Matyja, A. Kubik, and Z. Stanik, "The mems-based barometric altimeter inaccuracy and drift phenomenon," *Zeszyty Naukowe. Transport/Politechnika Śląska*, 2022.

About the Author



Dimitrios Mpouziotas received his B.Sc. degree in Computer Science (2018–2024) and his M.Sc. degree in Computer Science & Networks from the Department of Informatics and Telecommunications, University of Ioannina, Greece. His research focuses on Computer Vision, Remote Sensing, and Geolocation Estimation, encompassing ecosystem monitoring, wildlife tracking using UAVs, offline/real-time inference, and end-to-end system design utilizing Object Detection, Multi-Object Tracking (MOT), Semantic/Instance Segmentation, and Multi-Modal Sensor Fusion. He has conducted research at the Laboratory of Knowledge and Intelligent Computing (KIC Lab) and the Industrial Systems Institute (ISI) of the "Athena" Research Center. He is the author of 9 peer-reviewed academic publications across IEEE, MDPI, and Taylor & Francis venues. He has delivered research presentations at international conferences including the IEEE Smart World Congress (SWC 2026) in Rende, Italy, and three editions of SEEDA-CECNSM (2022, 2024, 2025)—as well as invited talks and public exhibitions, such as BEYOND 2025 in Athens, Researcher's Night in Xanthi, and the 10th & 11th Panhellenic Exhibitions of Arta. His recent work focuses on modifying and optimizing deep learning architectures (CNNs, Vision Transformers) for remote sensing and geospatial analysis, including the PMV-Geo framework for ultra-high-resolution geospatial mapping of coastal salt marshes in the Amvrakikos Gulf from UAV imagery, and decadal satellite-based monitoring of marsh health, accretion, and regression in the Venice Lagoon.